AD-A085 282   MITRE CORP BEDFORD MA   F/6 17/2
MODELS OF LOCAL COMPUTER NETWORKS.(U)
APR 80 C TROPPER   F19628-79-C-0001
UNCLASSIFIED MTR-3783   ESD-TR-80-111   NL

1 OF 2
AD A
085282

LEVEL II

ESD-TR-80-111

MTR-3783

MODELS OF LOCAL COMPUTER NETWORKS

ADA085282

CARL TROPPER

APRIL 1980

Prepared for

DEPUTY FOR TECHNICAL OPERATIONS
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
Hanscom Air Force Base, Massachusetts

Project No. 572C

Prepared by
THE MITRE CORPORATION
Bedford, Massachusetts

Contract No. F19628-79-C-0001

REVIEW AND APPROVAL

This technical report has been reviewed and is approved for publication.

DALE W. GILBERT, Capt, USAF
Computer Systems Engineer
Technology Applications Division

CHARLES J. GREWE, JR., Lt Colonel, USAF
Chief, Technology Applications Division

FOR THE COMMANDER

NORMAND MICHAUD, Colonel, USAF
Director, Computer Systems Engineering
Deputy for Technical Operations

(UNCLASSIFIED)

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

19

1. REPORT NUMBER: 18 ESD-TR-80-111
2. GOVT ACCESSION NO.: AD-A085 282
3. RECIPIENT'S CATALOG NUMBER:
4. TITLE (and Subtitle): 6 MODELS OF LOCAL COMPUTER NETWORKS.
5. TYPE OF REPORT & PERIOD COVERED: 9 Technical rept.
6. PERFORMING ORG. REPORT NUMBER: 14 MTR-3783
7. AUTHOR(s): 10 Carl Tropper
8. CONTRACT OR GRANT NUMBER(s): 15 F19628-79-C-0001
9. PERFORMING ORGANIZATION NAME AND ADDRESS: The MITRE Corporation, P.O. Box 208, Bedford, MA 01730
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: Project No. 572C
11. CONTROLLING OFFICE NAME AND ADDRESS: Deputy for Technical Operations, Electronic Systems Division, AFSC, Hanscom AFB, MA 01731
12. REPORT DATE: 11 APR[redacted]80
13. NUMBER OF PAGES: 152 12 154
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office):
15. SECURITY CLASS. (of this report): UNCLASSIFIED
15a. DECLASSIFICATION DOWNGRADING SCHEDULE:
16. DISTRIBUTION STATEMENT (of this Report): Approved for public release; distribution unlimited.
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report):
18. SUPPLEMENTARY NOTES:
19. KEY WORDS (Continue on reverse side if necessary and identify by block number):
   ANALYTIC SIMULATION MODELS
   LOCAL COMPUTER NETWORKS
   PERFORMANCE MODELS
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This paper presents a discussion of analytical and simulation models of locally distributed computer networks that make use of ring or bus technology.

The section of the paper devoted to ring architectures focuses on models of the Pierce, Newhall, and DLCN rings. The emphasis in the bus section of the paper is on the modeling of access protocols.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

235050 JOB

ACKNOWLEDGMENT

This report has been prepared by The MITRE Corporation under Project No. 572C. The contract is sponsored by the Electronic Systems Division, Air Force Systems Command, Hanscom Air Force Base, Massachusetts.

TABLE OF CONTENTS

TABLE OF CONTENTS (Concluded)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Continued)

LIST OF ILLUSTRATIONS (Concluded)

LIST OF TABLES

SECTION 1

INTRODUCTION

PURPOSE

This paper presents a discussion of computer models for locally distributed computer networks that use ring or bus technology to connect the nodes (e.g., computers, terminals) to one another. Much work has been done on the modeling of both ring and bus networks, but the results are scattered throughout the literature. Hence, it was felt that in order to have the capability of comparing the many design alternatives possible in the construction of a local network, a survey of the modeling of these networks should be undertaken. The results of this survey are presented in this paper.

An earlier paper discussed models used for design and analysis of the communications system employed for centralized and distributed computer networks [TROP78]. Both of these papers were written as part of Project 572C, C^3 System Performance Modeling and Simulation, which focuses on the use of modeling and simulation for system-level design and analysis during the conceptual phase of systems acquisition.

TECHNICAL BACKGROUND

From a functional point of view, a local computer network may be thought of as inhabiting a region between multiprocessing systems and the interconnection of geographically distributed, heterogeneous computer systems for the purpose of resource sharing. At one end of this spectrum we find an attempt to convert a collection of serial processors into parallel processors, while at the other end we discover a group of dissimilar computers tied together by a communication network, thereby enabling a user to take advantage of a variety of computing resources. Local computer networks may therefore be built to fulfill either (or conceivably both) of the above design goals.

Metcalfe, in "Ethernet: Distributed Packet Switching for Local Computer Networks," uses a taxonomy based on the parameters of bit rate and separation between computers to distinguish three types of networks [METC76, p. 395].

Activity	Separation	Bit Rate
Remote Networks	>10 km	<0.1 Mps
Local Networks	0.1 - 10 km	0.1 - 10 Mps
Multiprocessors	<0.1 km	>10 Mps

It should be pointed out that the introduction of fiber optics technology into this area threatens Metcalfe's taxonomy.

Another definition for local networks has been proposed by A. Franck [FRAN78]. He pictures a local computer network as consisting of three essential ingredients:

1. A high-speed transmission medium for data transmission over a "limited" distance. The nature of the transmission medium and the topology of the network are left unspecified.

2. Several network adapters attached to this transmission medium which serve as line interfaces for computing equipment. The adapters transmit data on the transmission medium.

3. Computing system components that can be attached to an adapter. Franck's illustration of his definition is shown in figure 1.

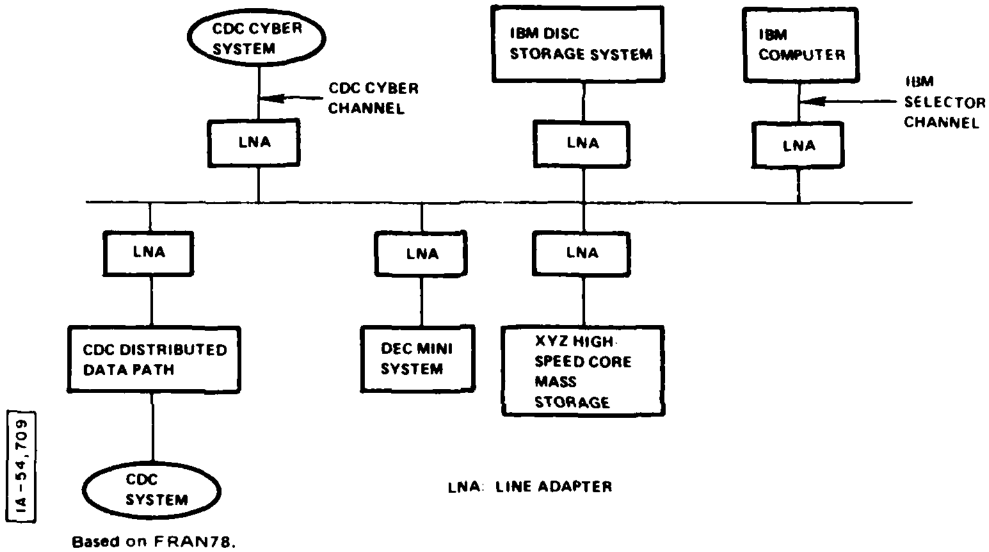


Figure 1. Local Computer Network

It should be noted that a panel discussion held during the third conference on local computer networks -- a conference devoted to developing a definition of a local computer network -- failed to achieve a definition acceptable to all participants.

Much effort has been devoted in recent years to developing technologies for local networks. Two of the basic networking technologies that have been developed are the ring and bus networks. In a ring network, the communications path is in the form of a loop.

Messages originate at a source node (attached to the loop) and then flow through intermediate nodes on the way to their destination node. The intermediate nodes on the path function as relays. In a bus network, on the other hand, messages are broadcast onto a shared communications channel. Hence, all nodes attached to the communications path "hear" all the messages sent onto the loop. More detailed definitions of these networks will be presented in sections 2 and 3.

To the author's knowledge, there have been no comparisons of the performance of ring and bus networks for a given application. This is unfortunate, as recent developments in ring technology (e.g., the DLCN ring, discussed in this paper) demonstrate that ring networks with impressive performance characteristics can be built.

SCOPE AND CONTENTS

As can be inferred from these brief definitions, the technologies employed in the construction of local networks vary widely. Hence, no truly general-purpose models for the design and analysis of local computer networks are available. Consequently, this paper will focus on the analytic and simulation techniques presently available for the synthesis/analysis of bus and ring networks.

Section 2 of this paper discusses models of ring networks. The section concludes with a summary of the strengths and weaknesses of the various models. An approach to the construction of models for ring networks is also presented. Section 3 is devoted to models of bus networks (including a taxonomy of bus networks). The final section of the paper contains several remarks on access protocols for bus networks and their modeling and includes a suggestion for constructing bus network models.

SECTION 2

RING NETWORK PERFORMANCE MODELS

INTRODUCTION

Following Anderson and Jenson, "Computer Interconnection Structures: Taxonomy, Characteristics, and Examples," [ANDE75], we characterize a ring network as a collection of processing elements (terminals or computers) that are interconnected via a communications path in the form of a loop.* This situation is illustrated in figure 2.

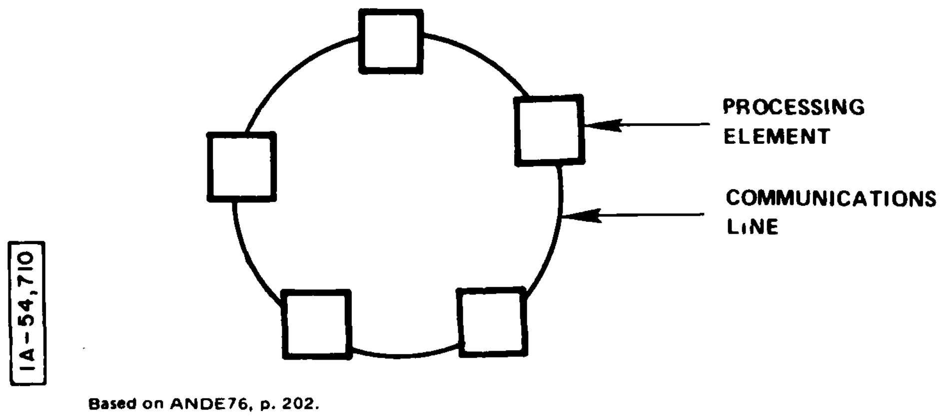


Figure 2. Ring Network

Typically, the processing element is attached to the ring by an interface device -- the ring interface unit. A loop supervisor may also be present on the loop. The functions of the supervisor may include synchronization, as well as some form of flow control (to prevent the accumulation of undeliverable messages).

* The words "ring" and "loop" are used interchangeably in the literature and in this paper.

In general, traffic on the loop flows in one direction only, although bidirectional systems have been proposed [MAJI76]. Hence, each processing element receives traffic from one of its neighbors, and sends messages to its other neighbor. Messages then circulate around the loop from their source to their destination. The intermediate processing elements along the path act as relays.

Depending on the system, messages may be of fixed or variable length, and one or several messages may be permitted on the loop at a time.

Loop networks are attractive because of their simplicity. It is fairly easy to add (or delete) processing elements without making numerous connections each time. This is a definite benefit when the network is located within the confines of an office building. In addition, start-up and system modification costs are (relatively) low.

The basic drawback of a loop system, its reliability, also stems from its simple design. An outage in either a processor or a channel can lead to disaster. Hence, it is necessary to provide some form of backup in the event of a failure. An example of such a backup would be installation of a bypass at each node, which would, in effect, delete a malfunctioning node. If the bypass is centrally activated (by a loop supervisor), it can also be used to route traffic around defective channels.

LOOP CONTROL ARCHITECTURES

Three major loop control architectures* have been developed: the Pierce loop, the Newhall loop, and the Distributed Loop Computer Network (DLCN). In the Pierce loop, fixed-length slots circulate around the ring. A lead field indicates to each host whether or not the next frame is

* The reader is referred to PENN78 for a comprehensive survey of loop architecture.

occupied. In the absence of a message, a host may multiplex a message (or a portion thereof) into the available slot. Figure 3 is a diagram of this transmission mechanism.

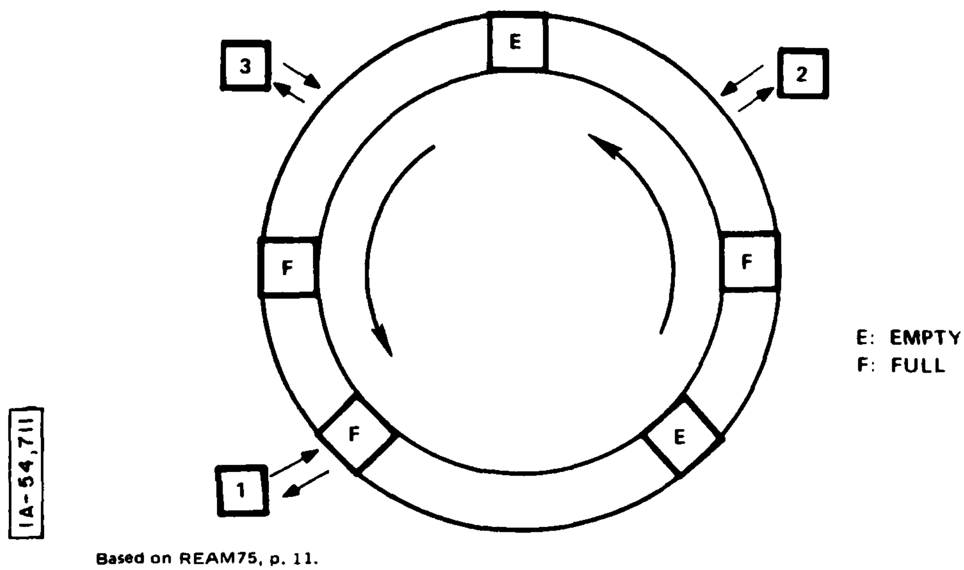


Figure 3. Pierce Loop Transmission Mechanism

Clearly, several messages may be sent on the network simultaneously. The principal disadvantage of this method is the fact that the messages are not all of uniform length. Some will be too short for the space allotted, resulting in a waste of space. Others will be longer than available space, necessitating software for assembly and disassembly of messages, as well as adequate buffer space.

The transmission of variable-length messages is a property exhibited by Newhall networks [FARM69]. These networks operate by "token-passing."

Control is passed from host to host. If a host receiving control of the loop has a message stored in its buffer, it immediately multiplexes the message onto the loop, and then passes control of the loop downstream. Clearly, simultaneous transmission of messages under these circumstances is impossible, because of message interference. Figure 4 illustrates the Newhall transmission mechanism.

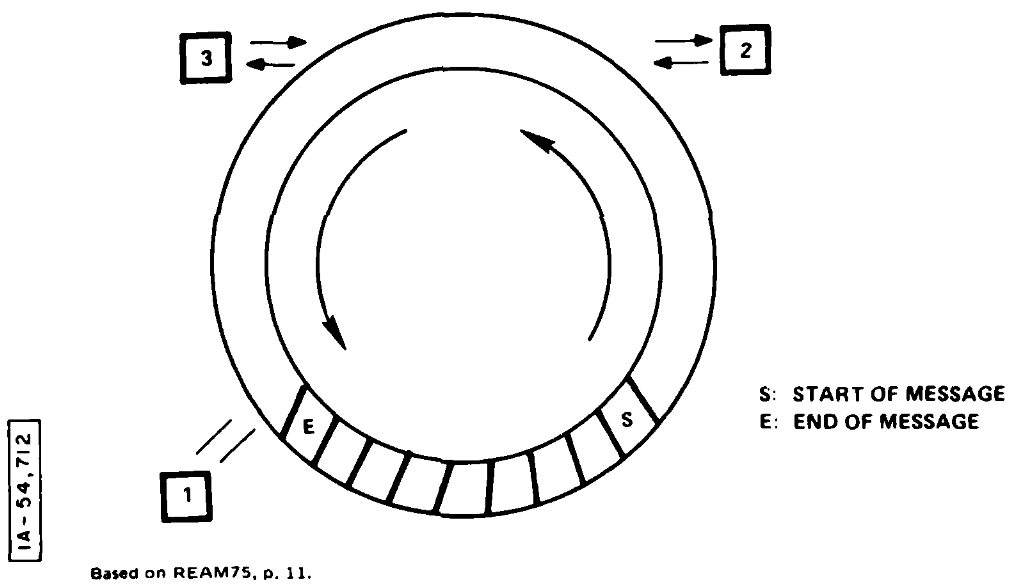


Figure 4. Newhall Loop Transmission Mechanism

The basic disadvantage of a Newhall network is, again, its inability to transmit several messages on the loop simultaneously.

DLCN provides both of these advantages (the advantages of variable-length messages and simultaneous message transmission) via a store-and-forward transmission. The ring-interface for DLCN consists of two buffers. The first is an output buffer which stores messages produced locally. The second is a delay buffer, which buffers messages passing through the particular node in question (i.e., messages that have destinations further downstream) and inserts messages from the output buffer into the gaps between messages on the loop, as well as into the gaps produced by sinking a message at the given node. The disadvantage of this approach is that delays occur for messages as the messages traverse nodes that lie on the path to their destination node.

MODELS

The following paragraphs describe models of the Pierce loop, Newhall loop, and DLCN architectures. Both simulation and analytic (queueing) models are discussed, but the emphasis will be on the queueing models, which will be of value in conceptual phase modeling. Results can be obtained quickly with queueing models, giving the system designer greater freedom to experiment with the system architecture. Another reason for the greater emphasis on queueing models is that, in the opinion of the author, they provide greater insight into significant design issues than do the more detailed simulation models. This is the result of the analyst being forced to carefully examine his assumptions in developing a mathematical model.

The DLCN model is described in more detail than the other two architectures because:

1. In the opinion of the author, the approach taken in this model -- modeling the nodes as an open Jacksonian network of queues, and incorporating this with a separate model of the communication loop subnetwork -- can also be employed in developing models of other loop systems.

2. The DLCN is capable of supporting both variable-length messages and simultaneous message transmission.

It should, therefore, have a wide range of uses in various applications. For example, plans have been made for it to support a distributed data base [PARD77].

A concluding section summarizes some of the limitations and advantages of the various models, and suggests a general approach to the modeling of loop networks.

The Pierce Loop

The transmission mechanism of the Pierce loop consists of multiplexing a message into one or more fixed-length time slots which circulate continuously around the ring. The loop was first proposed by Pierce [PIER72a] to accommodate a population of users which generate traffic characterized by a high peak-to-average ratio, i.e., "bursty" traffic. Inquiry-response systems, such as credit card verification and electronic funds-transfer, are examples of systems expected to support a bursty population. Pierce has also suggested the possibility of nationwide loop networks, which would consist of a large national loop attached to several regional loops, attached in turn to local loops to be used as the access mechanism to the network [PIER72b].

The performance of the Pierce loop has been discussed in papers by Hayes and Sherman, and Anderson et al. [HAYE71 and ANDE72]. Hayes and Sherman developed two analytical models of the system and compared their predictions to those produced by a GPSS* simulation. Details of this simulation and the results of the studies conducted with it form the contents of ANDE72.

* GPSS is an acronym for General Purpose Systems Simulator, a simulation language created by IBM.

The two models developed in HAYE71 were intended to complement each other. The first model portrays a population of bursty users, while the second focuses on sources that tend to generate longer messages. Both models depict a collection of data sources attached to the ring. The output of each of these data sources consists of alternating active and idle periods. It is assumed that the lengths of the active and idle periods are exponentially distributed and statistically independent of one another. Each message is then divided into a collection of fixed-length packets, and awaits its turn to be multiplexed onto the loop. Both models first develop an expression for the length of the busy and idle periods on the line, and then develop an expression for the average time delay based upon these calculations.

As has been stated, the first model portrays a collection of bursty users on the ring. By assumption, the lengths of the busy and idle periods at each individual source are exponentially distributed with known mean. It may be shown that the resultant idle period on the line (as seen by an arbitrary station on the ring) is also exponentially distributed. Its mean is equal to the sum of the mean values of all the "source" idle periods, i.e., of all the traffic generated by individual nodes feeding traffic into the chosen node. Using this method, the lengths of both the busy and idle periods can be calculated. To calculate time delay, Hayes and Sherman rely on a queueing model that views the line as a server subject to periodic breakdown [AVI63]. Line busy periods are interpreted as periods during which the server is active; idle periods are interpreted as breakdowns. The interested reader should consult HAYE71 for details of the calculations as well as for the expressions for time delay and line idle and busy periods.

The second model developed in HAYE71 is oriented toward more active sources on the line. As such it assumes that the mean length of the line idle period at the input to a particular station on the ring is known. A further assumption is that the data flow from a station is at a constant rate equal to the average rate. An expression is developed for computing the length of the idle and busy periods at the output of the station, and this expression is applied in iterative fashion around the ring. Based upon work by Sherman, the probability density function of the contents of the buffer at an arbitrary station is derived [SHER70]. An application of Little's theorem [KLEI75] yields an expression for the time delay.

Hayes and Sherman also developed a simulation model of the ring, which was run with 10, 50, and 100 stations on the ring at various line loadings. The simulation model assumed a symmetric traffic pattern (intended to represent the traffic situation that might be encountered in a national ring). The results obtained with the analytical models compared well with the simulation results (especially for moderate line loadings, below 0.5), except that the analytical models produced more conservative time delay values.

Newhall Loops

The transmission mechanism in the Newhall loop is characterized by token passing -- control of the loop is passed from host to host successively, with each host multiplexing its message onto the loop as it gains control. Farmer and Newhall, in their paper "An Experimental Distributed Switching System to Handle Bursty Computer Traffic," describe the initial design and implementation of the Newhall loop at Bell Telephone Laboratories, Holmdel, New Jersey [FARM69]. The original system consisted of several peripherals (Calcomp plotter, teletype, etc.) attached to the loop, along with a Honeywell 516 computer employed as loop supervisor. The reader should consult FARM69 for a detailed description of the system.

Following initial design, several models of the Newhall loop were developed. Two of these models determine the mean scan time of the loop, i.e., the time required for the token to pass around the loop. Average waiting times are also available, but are subject to restrictions (e.g., the terminal output buffer can contain at most one message). In fact, a comprehensive response-time model for the Newhall loop is absent from the literature.

In presenting the available results on the Newhall loop, therefore, we will first discuss scan time results, based on Yuen et al. [YUEN72] and Carsten et al. [CARS77], and then average message waiting time, as presented in KAYE72. As these three papers employ relatively simple probabilistic arguments, and they do not present a comprehensive response-time model of the Newhall loop, our remarks must be brief. We will devote somewhat more space to a discussion of Labetoulle et al., "A Homogeneous Computer Network" [LABE77]. This paper is interesting because in addition to portraying the loop itself, it portrays two host processors attached to the loop, and it models both the processors and the loop as networks of queues. (A good discussion of queueing networks may be found in KLEI75 and KLEI76.) The present author feels that networks of queues provide a fruitful approach to the modeling of ring networks. The importance of networks of queues as a modeling tool for ring networks is illustrated in this paper.

In the paper, "Traffic Flow in a Distributed Loop Switching System" [YUEN72], the system under consideration is a collection of (buffered) terminals, attached to a Newhall loop, which communicate with one another via fixed-length messages. Assuming Poisson input at each of the terminals, Yuen, et al. obtain results for the mean and variance of the scan time in the case of identical (symmetric) and nonidentical (asymmetric) input traffic at each of the terminals. A critical assumption for this model is that of light traffic conditions. This condition is expressed mathematically in the symmetric loop case by the inequality

$$\lambda NT_s << 1,$$

where λ = identical arrival rate,

N = number of terminals on the loop, and

T_s = service time for a terminal.

The expression obtained for the mean scan time T in the case of the symmetric loop is

$$T = \frac{NT_B}{1 - \lambda NT_s},$$

where T_B is the time delay due to the token passing (assumed to be one bit).

Other quantities of interest determined in this paper were formulas for the blocking probabilities at the terminals.

Simulation studies of the system were also conducted, and the results compared to the analytic results. As expected, the results were in close agreement for low traffic conditions, but mean scan time and blocking probability results diverged when the traffic became heavier.

In "A Simplified Analysis of Scan Times in an Asymmetrical Newhall Loop with Exhaustive Service" [CARS77], a number of terminals are also attached to the loop. The terminals are represented as having infinite buffer capacity, and the loop provides service for variable-length messages. As mentioned by the authors, the infinite buffer assumption is realistic because it is inexpensive to incorporate extra memory in a terminal. In addition, if one has host computers attached to a Newhall loop, there should be ample space in the host memory to alleviate any concern about buffer overflow.

Formulas for the mean and variance of the scan time are obtained under the usual assumption of Poisson arrivals. The formula for the mean scan time $E(t_s)$ is

$$E(t_s) = \frac{D}{1 - \rho} ,$$

where D = scan overhead (control character recognition, etc.),

ρ = loop utilization, given by $\rho = \frac{1}{\alpha} \sum_{i=1}^{N} \lambda_i$,

λ_i = message arrival rate (messages/sec),

α = line capacity (messages/sec).

Note the similarity between this formula and the usual time delay formula for an M/M/1 queue.

The formula for scan time variance is a bit more complicated. It depends upon the position on the loop from which one commences the scan. Consequently, in order to simplify the calculations approximate results which neglect the dependence for this quantity are also derived.

A four-node Newhall loop simulation produced results (mean scan time and variance) in close agreement with those obtained via the analytical model developed by Carsten et al.

In "Analysis of a Distributed Control Loop for Data Transmission," the author also considers a collection of terminals attached to a Newhall loop [KAYE72]. Each terminal is assumed to have a buffer containing exactly one (fixed-length) message, resulting in the loss of messages generated while the buffer is full. (As noted earlier, this is a somewhat unrealistic assumption.) Identical Poisson arrival rates are also assumed at each terminal.

In light of these assumptions, Kaye develops an expression for the distribution of the waiting time at a terminal, defined to be the time between the loading of a message into the terminal's buffer and the moment that its transmission commences. With this distribution in hand, expressions for the mean and variance of the waiting time are then readily obtained, as is an expression for the proportion of messages lost at a terminal during a scan. These expressions are too complex to be included herein; however, the interested reader may readily peruse them in KAYE72. Unfortunately, Kaye conducted no simulation to verify his results.

MININET

The MININET is a two-host network of minicomputers designed to support a distributed data base. An essential feature of the data base is that it can be partitioned into components, each of which will be queried by users located in a particular geographical region. MININET was developed to support transactions processing -- short queries followed by rapid responses. Credit card inquiries are an example of this sort of application.

In view of the expected bursty nature of the traffic, the system's designers chose to implement it in the form of a two-host Newhall loop. A description of the system and of modeling work done prior to its implementation may be found in LABE77. As the focus of the present paper is on system modeling, we urge any readers interested in details about the hardware, operating system, etc., to consult that paper. In the process of designing the network, both analytic and simulation models of the proposed system were developed. These models are also described in LABE77.

The queueing model developed in LABE77 is significant. It is the first attempt (that the author is aware of) to represent the entire network (i.e., host processors in addition to the loop communication subnetwork) as a network of queues*. This formulation was employed in deriving expressions for the response time and queue lengths at the nodes. Figure 5 shows the model of the MININET network.

As indicated previously, there are two host processors, connected to one another via a Newhall loop. Transactions enter the system (at a rate of λ/sec) from terminals attached to the hosts, and queue for access to the host CPU (represented in the diagram by FM, file machine). The command processors, terminal processors and message switch all reside at the FM. As transactions arrive, the FM determines whether or not the host CPU has the requisite data. In the event that it does, the FM routes the request to the data host (DH). The DH is a separate minicomputer, in charge of secondary storage. A transaction may require several accesses, as indicated by the arrow returning to the data host. After completing the requisite number of memory accesses, post-processing is performed in the FM, and the request is routed back to the originating terminal. In the event that a request must be satisfied remotely, it is routed onto the loop, and ultimately makes its way to the appropriate DH.

The service rates μ_1 through μ_6 are circled in the figure, while the number of items in the six queues are represented by the letters n_1 through n_6. The various branching probabilities are designated p_1 through p_4.

* This approach pointed toward a new and potentially very effective modeling technique for ring networks which is discussed in the section on DLCN.

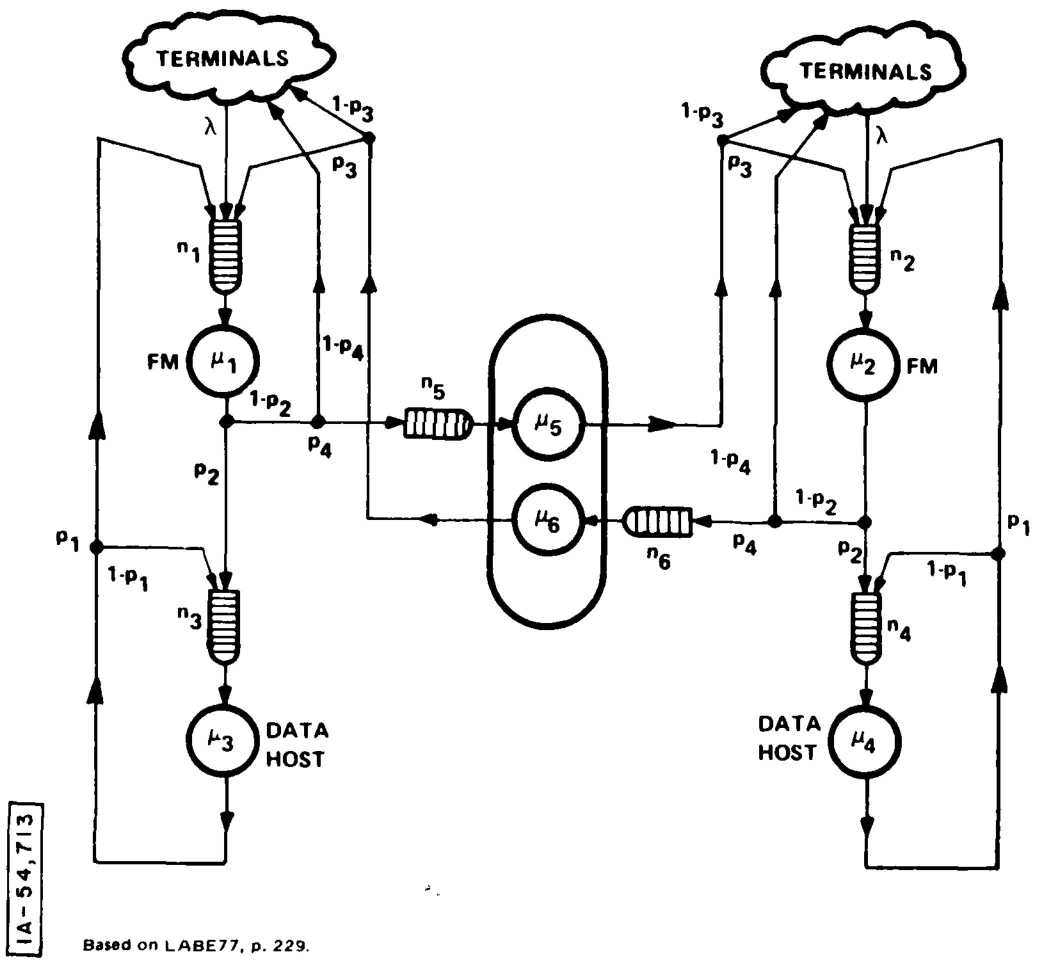


Figure 5. Model of MININET

Figure 5 represents what is referred to in the queueing literature as an open network of queues*, a collection of nodes (service centers), with customers visiting various centers guided by a matrix of node-to-node transition probabilities. The network is called "open" because customers are permitted to enter and exit from the system at the individual nodes.

Hence, the arrival rate at the i^{th} node, λ_i, may be written as

$$\lambda_i = \gamma_i + \sum_{j=1}^{N} r_{ji} \lambda_j,$$

where γ_i is the external arrival rate at node i, λ_j is the arrival rate at node j and the r_{ij} are transition probabilities.

Jackson established that for such a network, one might obtain the distribution for the number of customers in the system by multiplying the individual distributions at the nodes [JACK63]. This assumes a Poisson arrival rate at the nodes as well as exponential service times. Hence, if $p(k_1, \ldots, k_n)$ represents the probability of k_1 customers at node 1, k_2 customers at node 2, etc., an expression for $p(k_1, \ldots, k_n)$ (in our case) is provided by

$$p\left(k_1, \ldots, k_n\right) = \prod_{i=1}^{N} \left(1 - p_i\right) p_i^{k_i},$$

where $p_i = \dfrac{\lambda_i}{\mu_i}$, the utilization of the server.

---

* An excellent summary of networks of queues and their applicability to the modeling of computer-communications networks appears in KLEI75 and KLEI76. The interested reader is urged to consult these volumes for a survey of the state of the art in this area.

More general results can be found in MUNT72 and BASK75. Labetoulle et al. [LABE77] used these more general results (in particular, the concept of local balance) in deriving the above distribution, perhaps unnecessarily.

In LABE77, the arrival rates λ_i are computed by solving the following equation, which corresponds to the equation already presented for arrival rates:

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \\ \lambda_6 \end{bmatrix} = \begin{bmatrix} \lambda \\ \lambda \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & p_1 & 0 & 0 & p_3 \\ 0 & 0 & 0 & p_1 & p_3 & 0 \\ p_2 & 0 & (1-p_1) & 0 & 0 & 0 \\ 0 & 0 & 0 & (1-p_1) & 0 & 0 \\ (1-p_2)p_4 & 0 & 0 & 0 & 0 & 0 \\ 0 & (1-p_2)p_4 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \\ \lambda_6 \end{bmatrix}.$$

All of the above parameters can be obtained by assumption or experimentation, except for the loop service parameters, μ_5 and μ_6. (The estimated service rate of the file machine provides us with values for μ_1 and μ_2, for instance.) To obtain these parameters, the authors model the Newhall loop as follows:

Let μ_L = the line service rate.
If $n_6 = 0$, i.e., the corresponding port is idle, then $\mu_5 = \mu_L$.
If $n_6 \neq 0$, i.e., this port is not idle, then $\mu_5 = \frac{\mu_L}{2}$.

Hence, if q = the probability that the port with service rate μ_6 is idle, then

$$\mu_5 = q\mu_L + (1 - 1)\frac{\mu_L}{2} = \frac{\mu_L}{2}(1 + q).$$

To express q in terms of μ_L, the authors assume that the probability that the loop is idle (q^2) can be expressed as follows:

$$q^2 = 1 - \frac{2\lambda}{\mu_L}.$$

This corresponds to approximating the loop by an M/M/1 queue with a total arrival rate of 2λ. (In such a queue, the probability of the server being idle is $1 - \rho$, where ρ is the utilization of the server.)

Substituting this expression for q in the previous equation, we obtain

$$\mu_5 = \frac{\mu_L}{2}\left[1 + \left(1 - \frac{2\lambda}{\mu_L}\right)^{\frac{1}{2}}\right].$$

Expressions for response time and queue lengths at the various service centers can now be obtained by noting that in this (Jackson's) model, each service center behaves as an independent M/M/1 queue.

A simulation model of the network was also created, and its results were compared with the queueing model. Response time and queue lengths at various servers were the primary quantities of interest in this comparison. The program is an event-stepped simulation written in SIMSCRIPT. The model upon which it is based is the queueing network displayed earlier, combined with -- as the authors put it -- a number of "refinements" to reflect the simplifications inherent in the queueing model (e.g., message transfer protocols).

For two-host networks, the results of the queueing and simulation models are in close agreement. Agreement is especially good if the fraction of traffic at any node which goes remote is assumed to be less than 0.6. (This is within the operating range of the network.)

The queueing model was generalized to more than two hosts and the results were compared to the corresponding simulation output. In this case, serious divergence occurred when the fraction of remote traffic exceeded 0.4. This is to be expected, as the simple two-host model created in LABE77 does not admit to a straightforward realistic generalization. The literature does not contain a good model of the Newhall loop with more than two ports.

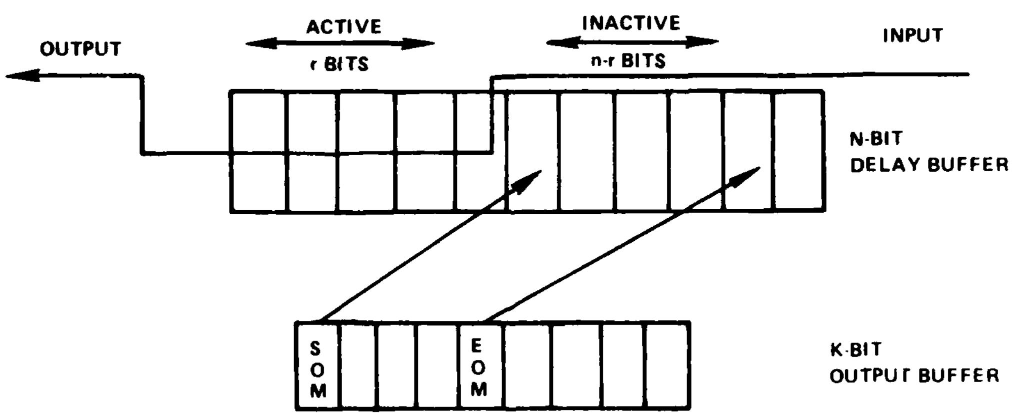
DLCN

The Distributed Loop Computer Network (DLCN) was designed and is presently being implemented at Ohio State University.* The design of this network is documented in a series of papers by Liu, Babic, Pardo, and Reames. Our description of the network and the associated queueing models is based upon Reames and Liu, "A Loop Network for Simultaneous Transmission of Variable-Length Messages" [REAM75], and Babic et al., "A Performance Study of the Distributed Loop Computer Network" [BABI77]. DLCN has been designed to provide:

1. Simultaneous message transmission between hosts.
2. Transmission of variable length messages.

* In a telephone conversation with the author, Professor Liu, who is associated with the project, indicated that he expects to have a prototype network (3 nodes) completed by the end of the summer.

The network thus combines the advantages of the Newhall network (item 2) and the Pierce network (item 1). DLCN provides both of these advantages via a store-and-forward solution. The ring-interface employed to effect this solution consists of two buffers, an output buffer which is used to store messages produced locally, and a delay buffer. The delay buffer stores messages passing through the particular node in question (i.e., messages that have destinations further downstream) and inserts messages from the output buffer into the gaps in between messages on the loop, as well as the gaps produced by sinking a message at the given node. A diagram of these two buffers (figure 6) will help explain their operation.



IA-54,714

Based on REAM75, p. 11.

Figure 6. Model of DLCN Ring Interface

We consider first the operation of the delay buffer. Bits arrive serially at the buffer along the incoming line, one per time unit. Assume that an on-going t-bit message arrives at times t_0, t_1, ... , t_{r-1}. These bits will be stored in the first r positions of the buffer, the active portion. As the last bit of this on-going message arrives, the first bit will be transferred onto the output line. In the event that another on-going message does not appear for another time unit, the active portion of the delay buffer is reduced by one bit. (Our first-in, first-out queue is reduced by one customer.) As the gaps between messages continue to appear, the size of the delay decreases -- approaching its irreducible minimum, which is one.

In the event a message of length s bits has been assembled in the output buffer and a gap appears between messages on the input line, the s-bit message is parallel-transferred into the inactive portion of the delay buffer immediately adjacent to the active portion. This transfer occurs under the provision <u>that there are at least s + 1 bits available in the inactive portion of the delay buffer</u>. The active portion of the delay buffer now contains r + s + 1 bits, while the remainder of the delay buffer now constitutes the inactive portion. The extra bit is used for delaying new, incoming messages. This is the conceptual mechanism employed for inserting messages into the gaps between messages on the input line.

In the event that the length of the message in the output buffer exceeds the space available in the delay buffer, it does not gain access to the delay buffer until the active portion is sufficiently reduced. This tactic clearly penalizes a heavy user of the system. However, the length of the delay buffer can be a design variable, capable of being increased to favor certain important users. The basic design trade-off brought to light is that of balancing the ability to output messages (achieved by employing smaller delay buffers).

A more detailed description of the operation of the ring interface transmitter, as well as several possible hardware implementations of the transmitter, may be found in REAM75.

Modeling of DLCN

Before the DLCN was implemented, both analytical and simulation studies of the network were conducted. We intend to focus on the queueing model because it provides a framework for the modeling of other ring networks (Pierce, Newhall, etc.). Comparisons of the analytic and simulation results will be included in our discussion. We therefore devote this section to a detailed summary of the model, based on papers by Babic et al. ("A Performance Study of the Distributed Loop Computer Network (DLCN)" [BABI77]) and Liu et al. ("Traffic Analysis of the Distributed Loop Computer Network (DLCN)" [LIU77]).

Two fundamental points about the DLCN queueing model are:

1. It portrays each node as an (open) Jacksonian network of queues. (A description of queueing networks may be found in KLEI76, pages 212 - 236.)

2. It approximates the loop communications subnetwork as a single-server queue.

Thus, the average time delay for the entire network can be calculated by computing the time delays at individual (host) nodes via a Jacksonian model, and determining the loop* time delay based upon the model described in LIU77.

For several reasons, the network was not modeled by simply appending queueing submodels for the hosts onto the model for the

* We risk some semantic confusion by writing "loop" for "loop communications subnetwork" in what follows.

loop subnetwork. The most important reason, in the opinion of the author, is that this approach would have resulted in an unwieldy number of equations. By employing an expression for the average subnetwork time delay, the authors greatly reduced the complexity of the model.

Two other difficulties that appear in the modeling of the loop subnetwork, which also mitigated against this approach, will be discussed in the next section. Briefly, they are:

- The loop service discipline alternates access priority to the loop between incoming messages and locally generated messages.

- A message is simultaneously serviced by more than one node.

Ultimately, the authors derive expressions for:

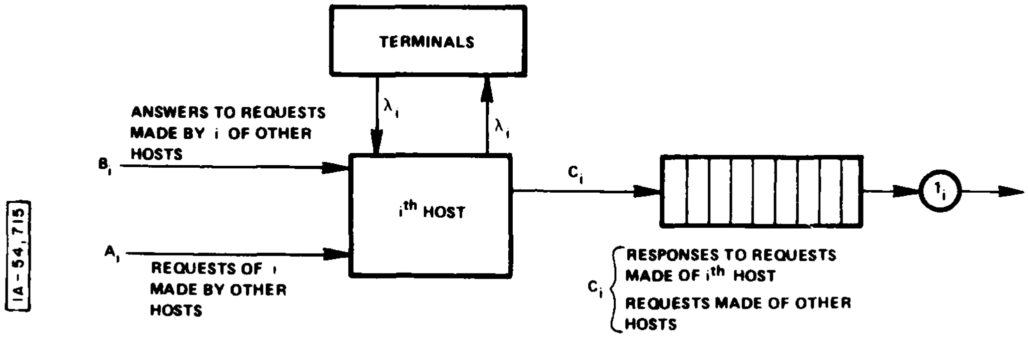
- channel and processor utilizations.
- queue lengths at the processors and at the channel.
- network time delay.

In describing their work, we first summarize the model for the entire network [BABI77] and then we summarize the model for determining the loop time delay formula [LIU77].

Conceptual Models of the DLCN

Figures 7, 8, and 9 [BABI77], depict the conceptual basis for modeling the DLCN. It is assumed that interarrival times and service times are distributed exponentially and that service follows a first-come, first-serve (FCFS) queueing discipline.

Figure 7 shows a simplified model of the host and its relationship to the network.



Based on BABI77, p. 74.

Figure 7. Model of Host Message Flows

As the figure shows, terminals that generate requests and receive responses at rate λ_i are attached to the i^{th} host. The message streams A_i, B_i, and C_i, by which the host interacts with the rest of the network, are also indicated.

Focusing on the host, we note that it is conceptually divided into a <u>communications server</u> and a <u>request server</u> (figure 8). The communications server receives requests from terminals, performs preprocessing functions on the requests and either routes them to the local host if they can be satisfied locally, or routes them to a remote host if they cannot. The communications server also receives both local and remote responses to requests, performs post-processing on them and returns them to the terminals that made the inquiries.

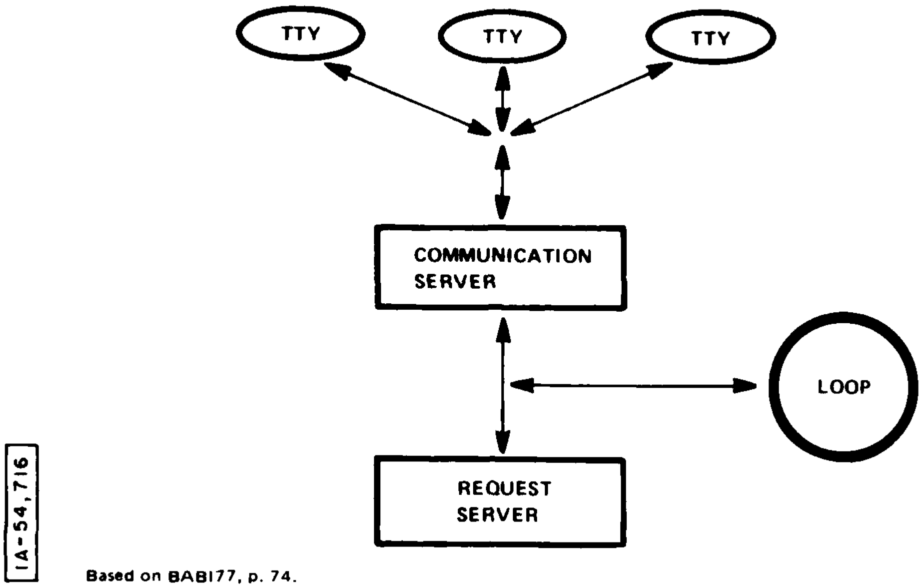


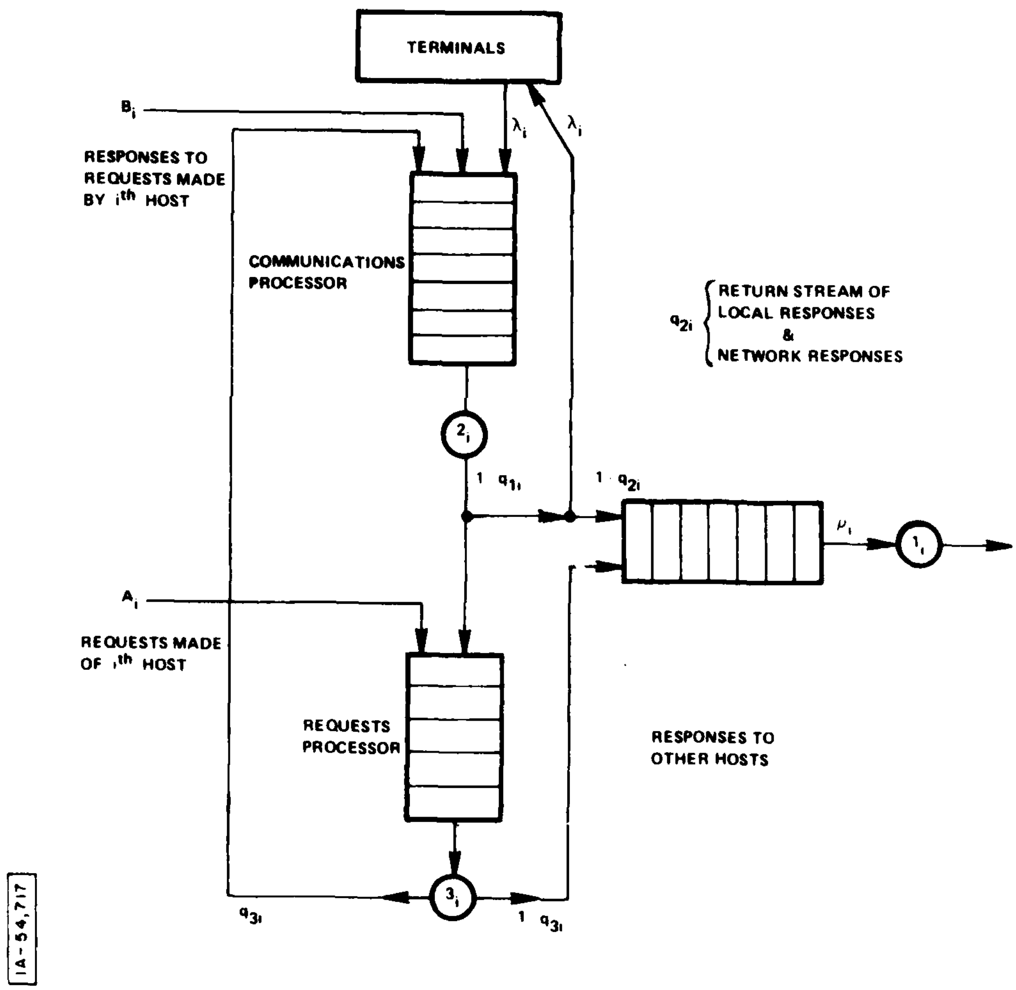
Figure 8. Model of i^{th} Host

Figure 9 shows a detailed flow of messages through the host and into the network.

In this figure, requests are seen entering the communications processor (at rate λ_i) from the terminals attached to the i^{th} host. Two other streams also seen entering the communications processor are:

1. Stream B_i, which consists of responses to (remote) requests made by the i^{th} host.

2. A return stream of responses from the request server of the i^{th} host.



Based on BAB177, p. 74.

Figure 9. Detailed Model of i^{th} Host

Both of these streams will undergo post-processing and will then be routed to the appropriate terminals. The communications processor is depicted as server 2, and processes requests at rate μ_{2i}. If the requests can be satisfied locally, they are passed onto the request server, represented as server 3. Th .equest server processes this stream at rate μ_{3i}. Stream A_i, consisting of requests made of the i^{th} host by remote hosts, competes with local requests for the attention of the request server.

In the event that local requests must be satisfied remotely, they are directed to the loop, which is represented as server 1, operating at rate μ_{1i}. This stream competes for the loop with responses to stream A_i (i.e., responses to requests made of the host). These two streams are merged to form stream C_i, and delivered to the loop. The remote responses to local requests are then returned (after processing) as part of stream B_i to the communications server, where they undergo post-processing. After post-processing, the responses are sent to the appropriate terminals.

The model described above fits neatly into the category of a Jacksonian open network of queues. Such a network consists of a collection of (N) nodes with associated queues, each of which behaves as an independent M/M/1 queue (i.e., exponential interarrival and service times with one server). The nodes in our case will correspond to host computers. The input rate, d_i, to node i may be calculated according to the following equation.

$$d_i = b_i + \sum_{j=1}^{N} r_{ji}\, d_j, \text{ for } i=1, \dots, N,$$

where

b_i = external arrival rate to node i. In our case, this corresponds to the terminals associated with the host.

r_{ji} = probability of a message being sent from node j to node i.

An informative summary of networks of queues may be found in KLEI75.

Calculation of Design Parameters

The following flow diagram (figure 10) indicates the approach described in BABI77 towards the calculation of design parameters.

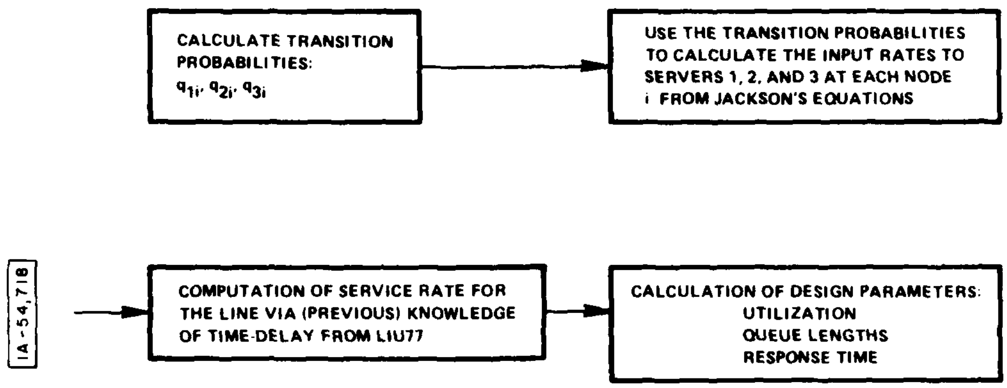


Figure 10. Calculation of Design Parameters

The following assumptions and definitions will be employed in our calculations:

- Assumptions

The following quantities are assumed known.

. It is assumed that a fractional traffic matrix F, defined as follows, is given.

$$F = \begin{vmatrix} 1 - \sum_{j=2}^{N} f_{ij} & f_{12} & \cdots & f_{1N} \\ f_{21} & 1 - \sum_{\substack{j=1 \\ j\neq 2}}^{N} f_{2j} & \cdots & f_{2N} \\ \vdots & \vdots & & \vdots \\ f_{N1} & f_{N2} & \cdots & 1 - \sum_{j=1}^{N-1} f_{Nj} \end{vmatrix}$$

The f_{ij}, i, j=1, ... ,N, represent the fraction of requests generated at the i^{th} host <u>which must be satisfied at the j^{th} host</u>.

- The Poisson arrival rate λ_i, (i=1, ... ,N) of requests from terminals associated with the i^{th} host, is assumed known.
- The exponential service rates μ_{2i} and μ_{3i}, (i=1, ... ,N) for the request and communication server (servers 2 and 3), are assumed known.
- The mean exponential message length $\frac{1}{\gamma_i}$ (i=1, ... ,N) is assumed known.

- <u>Definitions</u>

From these quantities, one can calculate the following parameters:

- F_1, (i=1, ... , N), the fraction of requests generated at the i^{th} host which are to be satisfied remotely.

$$F_i = \sum_{\substack{j=1 \\ j \neq i}}^{N} f_{ij}.$$

- σ_i, (i=1, ... , N), the arrival rate of requests to host i from remote sources:

$$\sigma_i = \sum_{\substack{k=1 \\ k \neq i}}^{N} f_{ki}\, \lambda_k.$$

With these definitions, the authors are able to proceed with the first state -- calculation of the transition probabilities.

They first note that the input to server 2 (request server) of the i^{th} host consists of the following four streams:

- Requests from terminals to be satisfied remotely, arriving at rate $F_i \lambda_i$.
- Requests from terminals to be satisfied locally, arriving at rate $(1 - F_i)\, \lambda_i$.
- Responses returning to remote requests made by host i at rate $F_i \lambda_i$.
- Responses to local requests returning at rate $(1 - F_i)\lambda_i$.

Referring to the detailed model of the i^{th} host, it is clear that only stream 2 gocs to the third server of host i. Hence,

$$q_{1i} = \frac{(1 - F_i)\lambda_i}{F_i\lambda_i + (1 - F_i)\lambda_i + F_i\lambda_i + (1 - F_i)\lambda_i} = \frac{1 - F_i}{2} .$$

To calculate q_{2i}, we note that

$$q_{2i} = \frac{\text{stream 2 + stream 3}}{\text{stream 1 + stream 2 + stream 3}}$$

$$= \frac{\left(1 - F_i\right)\lambda_i + F_i\lambda_i}{F_i\lambda_i + (1 - F_i)\lambda_i + F_i\lambda_i}$$

$$= \frac{1}{1 + F_i} ,$$

where we liberally interpret streams 1, 2, and 3 as the arrival rates of these streams.

To calculate q_{3i}, we note that the input to the request server (server 3) consists of stream 2 and stream A_i.

With the transition probabilities in hand, we now calculate the input rates to the three servers of the i^{th} host, d_{1i}, d_{2i}, and d_{3i} as follows:

$$\begin{bmatrix} d_{1i} \\ d_{2i} \\ d_{3i} \end{bmatrix} = \begin{bmatrix} 0 \\ (1 + F_i)\lambda_i \\ \sigma_i \end{bmatrix} + \begin{bmatrix} 0 & (1 - q_{1i})(1 - q_{2i}) & 1 - q_{3i} \\ 0 & 0 & q_{3i} \\ 0 & q_{1i} & 0 \end{bmatrix} \begin{bmatrix} d_{1i} \\ d_{2i} \\ d_{3i} \end{bmatrix} .$$

Solving the above set of equations, we obtain

$$d_{1i} = \sigma_i + F_i\lambda_i \ ,$$

$$d_{2i} = 2\lambda_i \ ,$$

$$d_{3i} = \sigma_i + (1 - F_i)\lambda_i \ .$$

This analysis assumes that the structure of each of the hosts is similar -- that each host consists of a request server and a communications front end. If the structure of each host differs, the number of equations will increase.

Computation of the Loop Service Rate

As was mentioned before, an analytical model of the time delay in the loop subnetwork was developed in LIU77. Using the results of this analysis, an expression for the service rate of the loop as seen by the i^{th} host can be obtained by substituting this expression in the formula for time delay in an M/M/1 queue [KLEI75, p. 98]. This yields the following expression for the loop service rate μ_{1i},

$$\mu_{1i} = \frac{1}{T_L} + d_{1i},$$

where T_L = time delay for the loop as derived in LIU77, d_{1i} = arrival rate.

Calculation of Parameters

The last box in figure 10 refers to calculation of the three fundamental design parameters:

- Utilization of request and communication processors as well as the loop communication channel.
- Queue lengths at each of these processors.
- Response time, defined as the time between the arrival of a request from the terminal at the local communication server and the delivery of a response to the terminal.

We approach these calculations as follows:

- Utilization. For processors 2 and 3 (request and communication server), the utilizations are calculated as follows:

$$U_{ij} = \frac{d_{ij}}{\mu_{ij}}$$

  for i=2, 3, and j=1, ... , N. The utilization of server 1 (loop) is calculated as in LIU77. The calculation will be described later.

- Queue Lengths. The average number of messages at the i^{th} server of the j^{th} host, N_{ij}, is given by

$$N_{ij} = \frac{d_{ij}}{\mu_{ij} - d_{ij}}$$

  for i=1, 2, 3, and j=1, ... , N.

- Response Time. The average queueing time, T_{ij}, at the i^{th} server of the j^{th} host is given by

$$T_{ij} = \frac{1}{\mu_{ij} - d_{ij}}$$

for i=2, 3, and j=1, ... , N. T_{1i} has been calculated before as T_L, for all i.

Using this formula, we can calculate the average response time for requests originating at the i^{th} host to be satisfied at the host denoted by T^{ij}.

The components of T^{ij} consist of the following five time delays:

- T_{2i}, the mean preprocessing time at the i^{th} host.
- T_L, the mean loop delay time to deliver the request from host i to host j.
- T_{3j}, the mean processing time at host j.
- T_L, for the return trip from j to i.
- T_{2i}, the mean post-processing time at host i.

Hence,

$$T^{ij} = 2\left(T_{2i} + T_L\right) + T_{3j}.$$

With this expression in mind, T^i, the average response time for remote requests from host i, may be calculated as follows:

$$T^{(i)} = \frac{1}{F_i} \sum_{\substack{j=1 \\ j \neq 1}}^{N} T^{(ij)} f_{ij}.$$

An expression for T, the average response time for any host, may also be obtained:

$$T = \frac{\sum_{i=1}^{N} F_i \lambda_i \; T^{(i)}}{\sum_{i=1}^{N} F_i \lambda_i} \; .$$

By similar reasoning, expressions for the average response time to satisfy a local request at the i^{th} host -- the system-wide response time for a local request -- may be obtained. Denoting these expressions by D^i and D respectively, we obtain

$$D^i = 2T_{2i} + T_{3i}$$

and

$$D = \frac{\sum_{i=1}^{N} \left(1 - F_i\right)\lambda_i \; D^{(i)}}{\sum_{i=1}^{N} \left(1 - F_i\right)\lambda_i} \; .$$

The average response time to satisfy any request from the i^{th} node, $TD^{(i)}$, is given by

$$TD^{(i)} = F_i \; T^{(i)} + \left(1 - F_i\right)D^{(i)} .$$

Hence the average system-wide response time is given by

$$TD = \frac{\sum_{i=1}^{N} \lambda_i \; TD^{(i)}}{\sum_{i=1}^{N} \lambda_i} \; .$$

These analytical results have been compared with results obtained from a GPSS simulation of the network. (Details may be found in REAM76.)

Six hosts were attached to the loop, each host in turn having 50 terminals attached to it. Times and message lengths were assumed to be exponentially distributed. The capacity of the communication channel was varied (10, 20, and 50 Kbps).

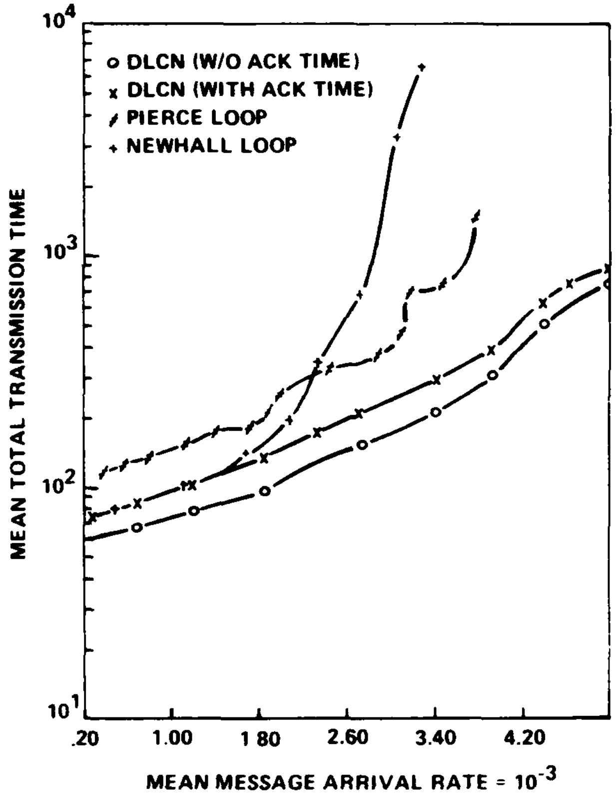
In order to compare the results of the analysis with the simulation, several parameters pertaining to loop performance were singled out and compared with the simulation results. The parameters were:

- processor and channel utilization.
- mean system response time and mean response time for a remote request.

Each of these parameters was plotted as a function of the capacity of the loop C and as a function of the fraction of requests to be satisfied remotely (already designated F). The channel and processor utilizations were reported to be in good agreement, as were the response times for high values of C. Some discrepancy did exist for lower values of C probably due to weakness in the formula for T_L.

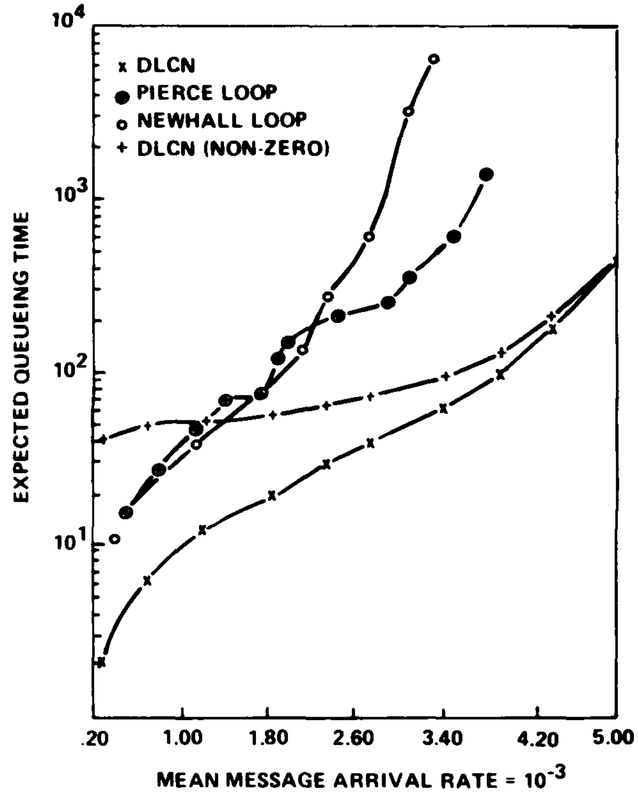
As part of the same simulation effort, a comparison was also made between the DLCN and the Pierce and Newhall networks [REAM76]. Two of the principal quantities of interest in the comparison were mean total transmission time, defined as time elapsed from message generation to removal of the last character of the message from the loop; and mean queueing time, defined as time elapsed between message generation and its placement on the loop. Figures 11 and 12 (reproduced from REAM76) portray the dependence of these two quantities upon mean message arrival rate. (A unit of time is equal to the amount of time required to transmit one character.) Both graphs portray the DLCN as superior to either the Pierce or Newhall networks.

IA-54,695



Based on REAM77, p. 128.

Figure 11. Mean Total Transmission Time for All Three Networks



Based on REAM77, p. 128.

Figure 12. Expected Queueing Time for All Three Networks

Model for Communications Loop

As mentioned in the preceding section, both analytical and simulation work was done to determine a reasonable expression for the time delay and channel utilization of the DLCN communications loop. The results are reported in LIU77. With this expression, an expression for the overall time delay for the network was obtained. This overall time delay is defined as the time between the arrival of a request from a terminal at the communications server, and the reception of a response at the terminal.

The conceptual model employed for the loop interface is shown in figure 13, reproduced from LIU77.

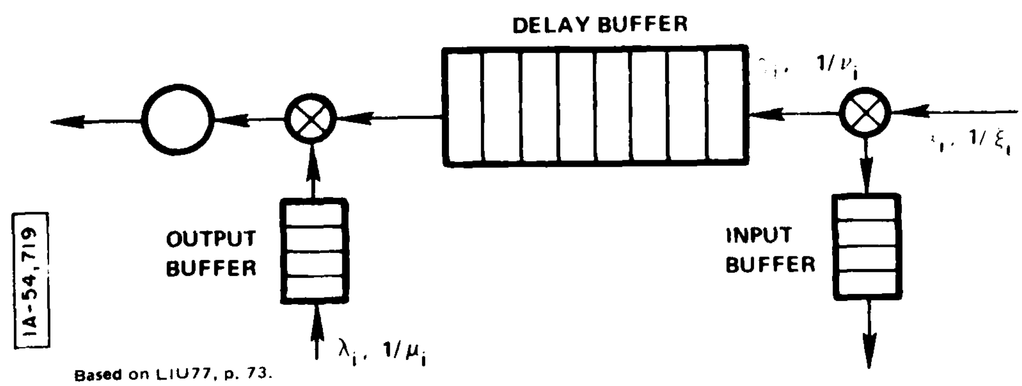


Figure 13. Loop Interface Conceptual Model

Three queues are depicted in this model, corresponding to:

1. The Output Buffer of the attached host, which contains messages with a mean arrival rate of λ_i/sec, and a mean message length of $1/\mu_i$.

2. The Delay Buffer, which receives messages arriving at a (mean) rate of γ_i with an average length of $1/\ \nu_i$.

3. The Input Buffer, which has parameters α_i and $1/\ \xi_i$.

The fundamental assumptions behind the model of the loop interface are:

1. Both the local data source and the message stream relayed from the input buffer are Poisson processes.

2. The message lengths are taken from a general distribution.

3. The interarrival times of messages and their lengths are independent.

Although neither of the last two assumptions are realistic, they are necessary to make the analysis tractable. The third assumption is the so-called "independence" assumption invoked by Kleinrock in his modeling of the ARPANET [KLEI76, pp. 321-22].

One of the fundamental difficulties in modeling the loop subnetwork is that it is a queueing system in which the priority alternates between two message streams -- the local input stream stored in the output buffer (queue 1), and the queue forming in the delay buffer (queue 2). As noted earlier, the incoming messages from the remainder of the loop have priority over locally generated messages until there is sufficient space in the delay buffer to accommodate a local message. At this point, priority switches to the output queue. Mathematically, this condition may be expressed as follows: Messages from queue 1 have priority over messages from queue 2 at time t if and only if the following equation holds at time t,

$$D_i - \sum_{j=1}^{k} m_j \geq S_i,$$

where

S_i = length of the first message in queue 1.

k = number of messages in the delay buffer.

m_j = length of the j^{th} message in the delay buffer at time t.

The loop subnetwork itself is modeled as a cyclic network of queues, as illustrated in Figure 14, also taken from LIU77.

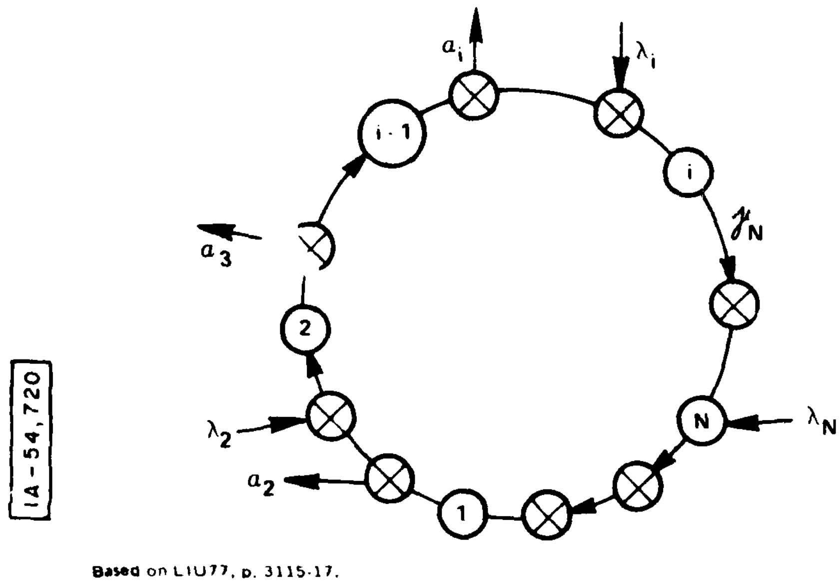


Figure 14. Loop Subnetwork Model

The numbers enclosed by circles represent channels. Preceding each channel, parameters indicating the average rate(s) of message arrival to and message deletion from the channel are shown. For example prior to channel i, messages are deleted at a rate of α_i and arrive at rate λ_i.

The second major difficulty in modeling the loop subnetwork is encountered at this point. Because of the almost instantaneous transmission speed of the loop, messages may be simultaneously served by two separate channels. For example, the set of characters that comprises a message from node i to node j may be partly in the output buffer preceding node i, and partly in the delay buffer of preceding node i + 1. Queueing theory unfortunately assumes that a customer may be served by only one server at a time. Liu et al. handle this problem by approximating the alternating nature of the loop service with a non-preemptive, head-of-the-line priority queueing system. This will be discussed at greater length.

Figure 15 presents an overview of the method set forth in LIU77 for calculating the parameters of network time delay and channel utilization.

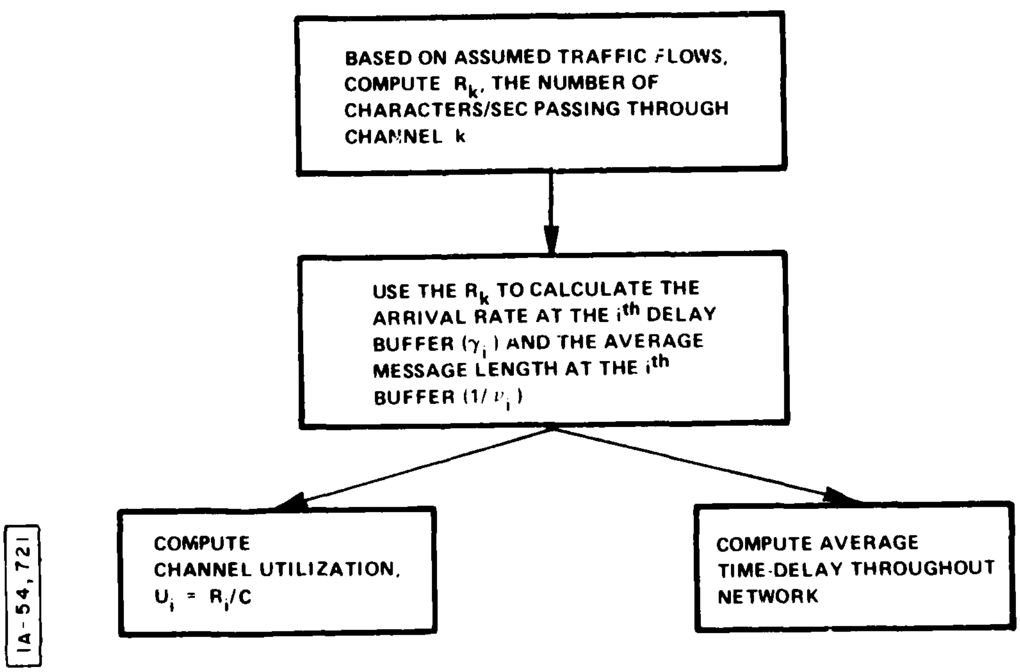


Figure 15. Flowchart for Parameter Calculation

Starting at the top of our flow chart, we assume that a traffic matrix P_{ij} is given. Each entry in the matrix represents that traffic originating at node i and destined for node j. P_{ii} is assumed equal to zero, and $\sum_{j=1}^{N} P_{ij} = 1$.

Our first step is to calculate R_{ik}, the average number of characters/sec originating* at node i and passing through node k (on the way to node j, for example).

The average number of characters/sec going from node i to node j is given by $P_{ij}\ \lambda_i/\mu_i$, as the input process is Poisson by assumption.

Hence an expression for R_{ik} can be obtained as follows:

$$R_{ik} = \begin{cases} \dfrac{\lambda_i}{\mu_i} \displaystyle\sum_{j=1}^{i-1} P_{ij} + \dfrac{\lambda_i}{\mu_i}, & 1 < i < k \neq N. \\[2ex] \dfrac{\lambda_i}{\mu_i} \displaystyle\sum_{j=1}^{i-1} P_{ij}, & 1 < i < k = N. \\[2ex] \dfrac{\lambda_i}{\mu_i} \displaystyle\sum_{j=k+1}^{N} P_{ij}, & 1 = i < k \neq N. \\[2ex] \dfrac{\lambda_i}{\mu_i} \displaystyle\sum_{j=k+1}^{i-1} P_{ij}, & k+1 < i \leq N. \\[2ex] \dfrac{\lambda_i}{\mu_i} & i = k. \\ 0 & \text{Otherwise.} \end{cases}$$

---

* Node i represents <u>channel</u> i, and traffic originates at a computer system attached to channel i.

These expressions are best understood by referring to figure 14.

R_k can now be calculated as

$$R_k = \sum_{i=1}^{N} R_{ik}, \quad k=1, \ldots , N.$$

Proceeding to the second box in figure 15, we calculate the parameters for the delay buffer, γ_i and $1/\nu_i$, as follows. (These exceptions are useful in calculating the average message delay.)

$$\gamma_i = \sum_{\substack{j=1 \\ j\neq i}}^{N} \mu_j \; R_{ji}.$$

This expression follows from the definition of R_{ji} as equal to $P_{ji}\ \lambda_j/\mu_j$, since $\mu_j\ P_{ji}\ \lambda_j/\mu_j = \lambda_j\ P_{ji}$, and the summation of these expressions must total to γ_i, the input to the delay buffer.

To calculate $1/\gamma_i$, the authors first define 1_{ji} to be that part of the traffic passing through delay buffer i which originates at node j.

Then,

$$1_{ji} = \begin{cases} R_{ji}/\left(R_i - \lambda_i/\,\mu_i\right) & \text{for } j\neq i, \\ 0 & \text{for } j=i \end{cases}$$

and

$$1/\gamma_i = \sum_{j=1}^{N} \left(1_{ji}/\mu_j\right).$$

Thus, $1/\gamma_i$ is a traffic-weighted average message length.

Proceeding to the third level of figure 15, we calculate the channel utilization at node i via the equation

$$U_i = R_i/C.$$

Finally, we calculate the average message delay in the network. Our first step is to obtain an expression for the average time delay T_{ij} between nodes i and j. There are five components in such a delay:

1. Waiting time in the delay buffer (queue 1), denoted by $T_1^{(i)}$.
2. Time required to multiplex the message onto the loop, denoted by $T_2^{(i)}$. This time is simply equal to M/C where M is the length of the message, and C is the capacity of the line.
3. Time T_3 required to check the address field of the message header at each intermediate node, equal to B/C seconds, where B is the number of characters in the address field.
4. Waiting time in each of the k intermediate delay buffers, denoted by $T_4^{(i + k)}$.
5. Propagation delay for the network, T_5. T_5 is negligible for a local network, and is therefore set equal to zero ($T_5 = 0$).

If we assume that there are r - 1 intermediate nodes between node i, the source, and node j, the sink, then we have the following expression for T_{ij}:

$$
\begin{aligned}
T_{ij} &= T_1^{(i)} + T_2^{(i)} + rT_3 + \sum_{k=1}^{r-1} T_4^{(i + k)} \\
&= T_1^{(i)} + M/C + rB/C + \sum_{k=1}^{r-1} T_4^{(i + k)} .
\end{aligned}
$$

To find the average message delay, we take expectations of both sides of this equation, and arrive at the following formula:

$$E(T_{ij}) = E(T_1^{(i)}) + {}^{1}/\mu_i C + rB/C + \sum_{k=1}^{r-1} E(T_4^{(i+k)}).$$

Note that $E(m) = \frac{1}{\mu_i}$ in this formula.

To get an expression for the average time-delay T in the network, we apply Little's law [KLEI75, p. 17] to the above formula, and obtain

$$T = \sum_{i=1}^{N} \left[E(T_1^{(i)})\ \lambda_i + \lambda_i/\mu_i C + E(T_4^{(i)})\ \gamma_i \right] \Big/ \sum_{i=1}^{N} \lambda_i$$

$$+ E(r)\ B/C\ ,$$

where

$E(r)$ = average path length, an expression for which is

$$E(r) = \sum_{i=1}^{N} \left(\gamma_i + \lambda_i\right) \Big/ \sum_{i=1}^{N} \lambda_i\ .$$

(See KLEI76, pp. 119-28.)

Expressions for $E(T_1^{(i)})$ and $E(T_4^{(i)})$ are missing from this discussion. It is in providing expressions for these two expectations that Liu and his associates provide their approach to the problem of the alternating priorities of their queueing structure. They assume that locally-generated messages always have priority over incoming messages, a situation that exists in low-traffic conditions (if the size of the delay buffer is large enough to hold any message generated at the node).

Given the assumption, the queueing structure fits neatly into the category of a nonpreemptive, head-of-the-line priority queueing system [KLEI76, pp. 119-123], for which the following formulas apply:

$$E(T_1^{(i)}) = \frac{W_0}{(1 - \rho_2)}$$

$$E(T_4^{(i)}) = \frac{W_0}{(1 - \rho_1)(1 - \rho_2)} ,$$

where

$$W_0 = \frac{\lambda_i \overline{a_i^2}}{2C^2} + \gamma_i \frac{\overline{d_i^2}}{2C^2} ,$$

$$\rho_1 = \frac{\lambda_i}{\mu_i C} + \frac{\gamma_i}{\nu_i C}, \text{ and}$$

$$\rho_2 = \frac{\lambda_i}{\mu_i C}.$$

A GPSS simulation was written to verify the assumptions of the analytic model, the primary quantities of interest being channel utilization and average message delay. The results of the simulation indicated good agreement on channel utilization. Under low traffic conditions (corresponding to a utilization of at most 0.3 or 0.4), the average message delays are also close. As traffic increases, agreement decreases as the analytic models provide more conservative results for the time delays. As mentioned by the authors, Liu et al., this discrepancy is no doubt due to:

- Assumption of Poisson arrival rate at the delay buffer
- Assumption of independence of message lengths and message interarrival times
- Approximation of the alternating priority structure by a fixed structure.

SUMMARY

This (ring) section has emphasized discussion of queueing models for the various major network architectures -- Newhall, Pierce, and DLCN. With but one exception [KAYE72], the queueing models were compared to simulations of the system under consideration. Agreement between the queueing model and the simulation was generally good, especially under conditions of moderate line loading. (The general rule-of-thumb definition for "moderate" is 60% utilization.)

Unfortunately, there appears to be no literature comparing these models (simulation or analytic) with the actual systems. People associated with the DLCN project say they intend to make such comparisons in the near future. It is certainly unfortunate that so little work has been done in comparing models and actual systems as much could be gained from such work.

It appears that the major difficulty in constructing queueing models of ring networks is the modeling of the ring subnetwork. The ring subnetwork must be distinguished from the system, which includes both the ring and the host processors attached to it. This has been a difficulty with all the models considered.

There is no comprehensive time-delay model for the Newhall network. A distribution for time delay has been derived under light traffic conditions [KAYE72], and a time-delay formula developed for the two-host case [LABE77]. Other results on the network are devoted to scan-time.

In modeling the Pierce network, Hayes was forced to construct two separate models for different traffic patterns [HAYE71].

Finally, the modeling of the DLCN encountered difficulties because of the alternating priorities of the queueing network and the fact that a message could be serviced simultaneously by two separate channels.

In light of all these difficulties, it appears that a reasonable approach to the modeling of ring networks would be to simulate the ring subnetwork, and to employ a (Jacksonian) open network of queues to represent the nodes. If heavy traffic conditions are not deemed vital to investigate, then the queueing models already discussed should prove to be adequate. As they provide conservative estimates for the time delay under heavy traffic conditions, little will be lost by employing them. In any event, further research in the development of queueing models for ring networks would certainly be of value.

SECTION 3

BUS NETWORKS

Using a bus as the communications subnetwork is a very popular approach to construction of local networks. G. Anderson and E. Jensen, in an article for Computing Survey [ANDE75], have created a taxonomy of the possible communications systems for a network of computers. They characterize a bus network as having its processing elements (computers, peripherals) attached to a common channel (the bus). Figure 16 shows their conception of a bus network.

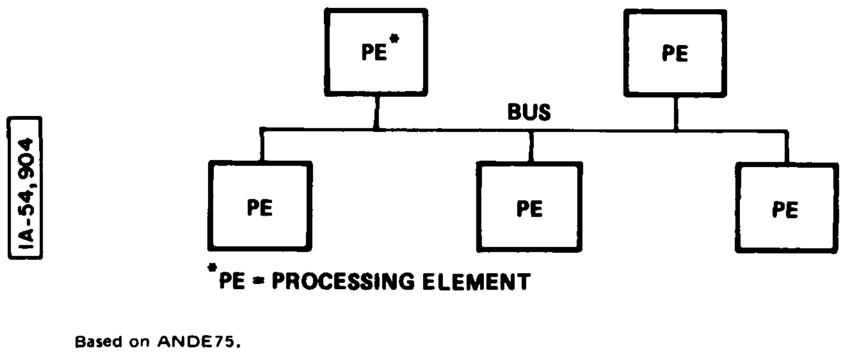


Figure 16. Bus Network

The channel itself is employed in a broadcast mode -- all processing elements "hear" a message. The transmission medium for the channel can be coaxial cable, optical fibers, twisted pairs, etc. Access to the network is controlled via some time-multiplexing technique.

As with ring networks, the major advantage of a bus network is its simplicity. It is easy to add or delete processing elements since numerous connections do not have to be made with each new addition (or deletion) of an element. In addition, start-up and modification costs are low, compared to other types of networks.

The principal disadvantage of a bus network is the vulnerability of the network to a failure of the bus itself. However, processing element failures are not catastrophic. A ring network must overcome this problem by using bypass units at the nodes.

In the case of either a ring or a bus network, some form of redundancy in the communication channel is necessary to eliminate the vulnerability of the system to a channel outage.

Performance evaluations of bus networks have centered on evaluating the access protocols by which the nodes gain access to the bus. In LUCZ78, E.C. Luczak develops an exhaustive classification of techniques employed in constructing bus networks. The bulk of his paper is devoted, in fact, to a description of channel access techniques.

Luczak points out that there are three major categories of access protocols: selection, random access (or contention), and reservation. We begin with a brief description of each of these categories (based on LUCZ78) and then proceed with a discussion of models of access protocols.

CHANNEL ACCESS TECHNIQUES

Selection

Selection techniques are the oldest access protocols. Their use originated in controlling access to multipoint communications lines as well as computer buses. The essential feature of a selection technique is that each node on the network must receive permission to send its messages onto the network. Until it receives this permission, it must queue its messages.

Selection techniques may be centralized or decentralized (distributed). In the case of centralized control, a central channel controller grants this permission, while in the distributed case, control is distributed throughout the nodes. The three types of centralized selection techniques are daisy-chaining, polling, and independent requests.

Daisy-chaining is a technique employed on internal computer buses. The PDP11 Unibus and the IBM 370 I/O channel both employ this technique. The signal line is "daisy-chained" through the nodes for the purpose of selecting the bus master. When a signal reaches a node, it can become bus master by stopping the signal and then broadcasting its messages. In the event that its message queue is empty, it simply propagates the signal onto the next node.

Both the advantages and the disadvantages of daisy-chaining arise from its simplicity. On the positive side, it is an easy technique to implement. On the negative side:

- It automatically imposes a fixed-priority structure.
- Its select pulses occasion time delays.
- It is vulnerable to failures in the grant line as well as the nodal interface.

In polling, a node is selected by being addressed. All nodes are informed of the next node to be selected. As in daisy-chaining, a node has the option of becoming bus-master or of refusing control of the line if that node does not wish to send any messages. The central controller may question each node in turn, or -- in the event of a prioritized nodal structure -- it may question them in a sequence determined by the priorities. This form of polling is often called roll-call polling.

Polling may be implemented on any serial channel, and therefore (unlike daisy-chaining) requires no special grant lines.

In the independent requests access technique, each node requests control of the bus from the central controller. The controller then ranks the requests according to their priorities, and selects the nodes accordingly. This system may be implemented in a variety of ways. Separate requests as well as separate select lines are one possibility on a parallel bus. Various time-multiplexing techniques may be employed on a serial channel for both requests and selections.

On a parallel bus, the independent requests method provides an efficient approach to the use of dynamic priority schemes. The major penalty to be paid, however, is the large number of control lines required for such a system.

Decentralized selection techniques that correspond to the three forms of centralized control have also been implemented (e.g., decentralized polling). A description of these techniques may be found in LUCZ78.

Random Access

Random access techniques are characterized by a lack of strict ordering of the nodes contending for access to the channel. In a random access technique, a node is free to broadcast its messages at a time determined by the node without being absolutely certain that no other node is simultaneously attempting to broadcast. In fact, in the original implementation of a random access protocol (the ALOHA system at the University of Hawaii), a node was free to broadcast whenever it had messages to send.

There is a great deal of interest in random access techniques for bus networks, primarily because of the bursty nature of computer traffic [FCH70]. A random access protocol provides the entire bandwidth of the channel to a user, once he gains access to the channel. Thus, if a small population contends for a channel at any given instant, a user with a message to transmit is guaranteed access to the full bandwidth of the channel after a brief waiting period.

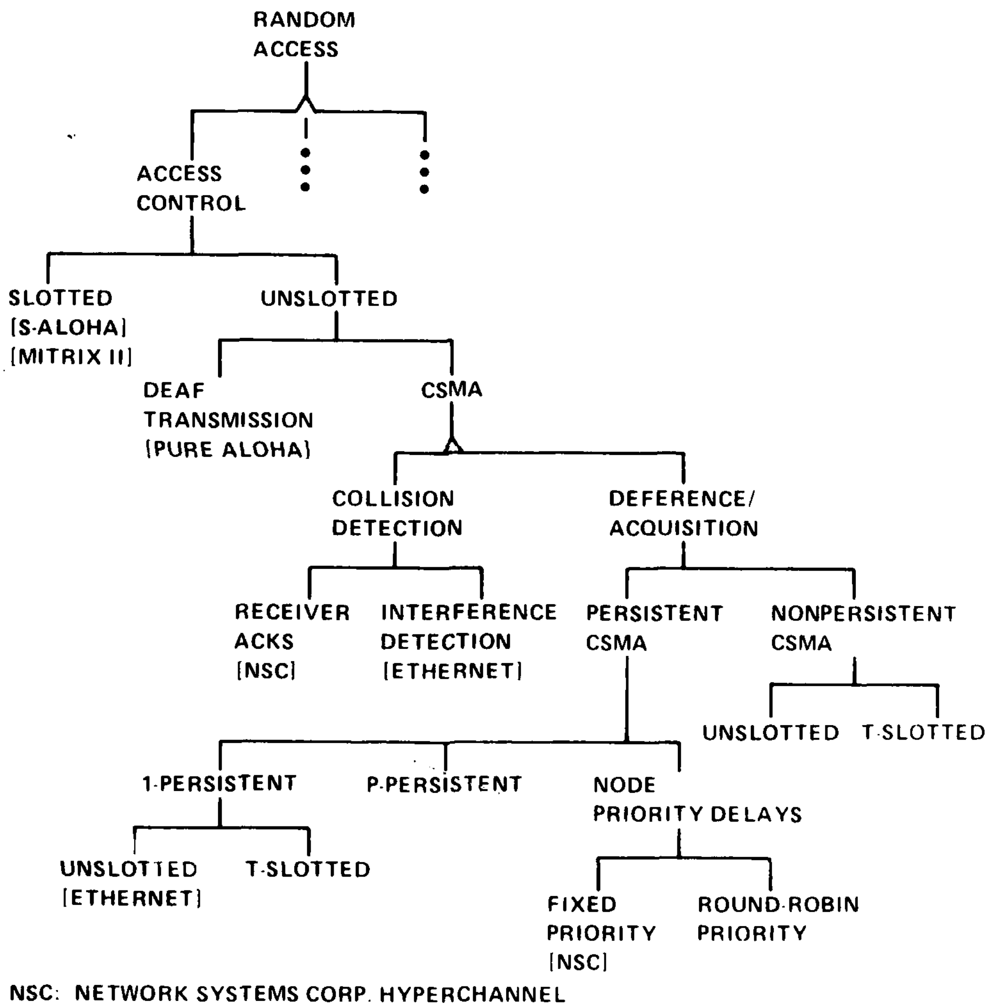
The price to be paid for random access is that messages may collide in transit. Collisions generally result in messages being rendered unintelligible. Hence, techniques have been developed for limiting access to the bus (thereby reducing the number of collisions that may occur) as well as for providing a retransmission sequence for messages that have collided (collision resolution).

This section continues with a description of random access techniques because of their importance and the great activity in this area. In presenting a discussion of random access techniques, we rely upon Luczak's taxonomy of random access protocols. Figure 17 presents Luczak's tree diagram of access control methods. Examples of systems employing those access methods are indicated on the diagram.

At the second level, random access techniques are either slotted or unslotted. In slotted techniques, all nodes are synchronized to a master clock. Time is subdivided into a collection of equal intervals. When a node has a message to send, it first subdivides the messages into packets of equal length (corresponding to the length of the time interval), which are then broadcast into the slots. If an acknowledgment for the packet is not received after some fixed period of time,*

*In a dual, unidirectional channel, an acknowledgment may be obtained by listening to the receiving channel. The signal should arrive after one propagation delay, thereby providing an automatic acknowledgment.

IA-54,895



Based on LUCZ78, p. 70.

Figure 17. Random Access Techniques -- Access Control

then the packet is assumed to have been destroyed by a collision, and is rebroadcast onto the network. The ALOHA technique is the epitome of random access techniques -- if a node has a message to transmit, it simply does so. To improve its efficiency, a slotted version was developed (referred to as slotted ALOHA).

Unslotted techniques permit the transmission of variable-length messages. As indicated by figure 17, there are two major categories of unslotted messages:

- Pure ALOHA or "deaf" transmission

- Carrier Sense Multiple Access (CSMA) techniques.

In CSMA, the sending node listens to the channel (senses carrier) before (and possibly during) message transmission. If carrier is sensed, the transmission is postponed for some period of time. If carrier is not sensed, the node is not guaranteed that its message will arrive safely. The message is still vulnerable for a fraction of the time required to transmit it onto the network. This fraction of time corresponds to one propagation delay. Since the propagation delay on a local bus network is considerably less than the transmission delay, the period of message vulnerability is brief. The objective of all CSMA protocols is to minimize the number of collisions that occur during this time window.

There are two fundamental facets of a CSMA protocol -- the collision detection method employed and the channel access technique employed (referred to as the deference/acquisition technique in Figure 17). Collision detection, as the name implies, is the means by which a node discovers that a message which it has sent has collided with another message. This can be accomplished by an acknowledgment broadcast by the receiving node (discussed previously) or by the simple approach of having the node listen to the channel for a short time subsequent to message propagation.* This idea forms the basis of the Listen-While-Talk protocol implemented on the MITRE bus system. When two or more users detect a collision, they immediately stop transmitting their message.

In order to understand how a node acquires control of the channel, one must know what the node does when the channel is sensed busy (deference), as well as what it does when the channel is sensed idle (acquisition). We will discuss these access techniques first. The two major access techniques in a CSMA protocol are persistent and nonpersistent CSMA. A nonpersistent CSMA protocol may be described as follows:

> A "ready" node (i.e., one with a message to transmit) senses the channel. Then
>
> 1. If the channel is idle, the ready node transmits its message.
>
> 2. If the channel is busy, it reschedules the message according to its collision resolution algorithm. (Typically, it picks a value out of a retransmission delay distribution.) It then repeats step one after the expiration of this delay.

The node is called nonpersistent because it does not continue to sense the channel after it has determined the channel is busy.

A defect of this approach is that several nodes that have already been involved in a collision might not broadcast on an idle channel if their retransmission delays have not expired. This defect led to the introduction of persistent protocols, in which a node continues to

*Twice the message propagation delay suffices to discover a collision.

sense a busy channel until it becomes idle, and then broadcasts its message. The obvious defect of such a protocol is that if two nodes have messages ready to transmit when the channel is busy, the messages will certainly collide when the channel becomes idle. To remedy this defect, a category of protocols with random transmission delays was introduced. Such protocols are called p-persistent protocols.

In a p-persistent protocol, time is divided into slots of length equal to the maximum propagation delay.

1. If a channel is sensed idle, it broadcasts a packet with probability p, and delays one slot with probability (1 - p), at which point it again senses the channel. If the channel is sensed busy, it waits until the channel is sensed idle, and repeats this step.

2. If the channel is again sensed idle in the next slot, it repeats the first step.

Thus, the p-persistent protocols represent an attempt to use all idle channels as soon as possible and at the same time decrease the number of collisions such immediate use might cause.

The probability of transmission (p) is a small number -- typical values are 0.03 and 0.1.

The last category of persistent CSMA protocols is built around the concept of nodal priorities. Here the idea is to define delay times based on priorities assigned to the nodes. If the channel is sensed idle, a ready node broadcasts its message. If it is sensed busy, the nodes delay an amount of time determined by their priorities. A sequence of delays, d_1, d_2, ... , d_N (N is the number of nodes) determines the order of broadcast.

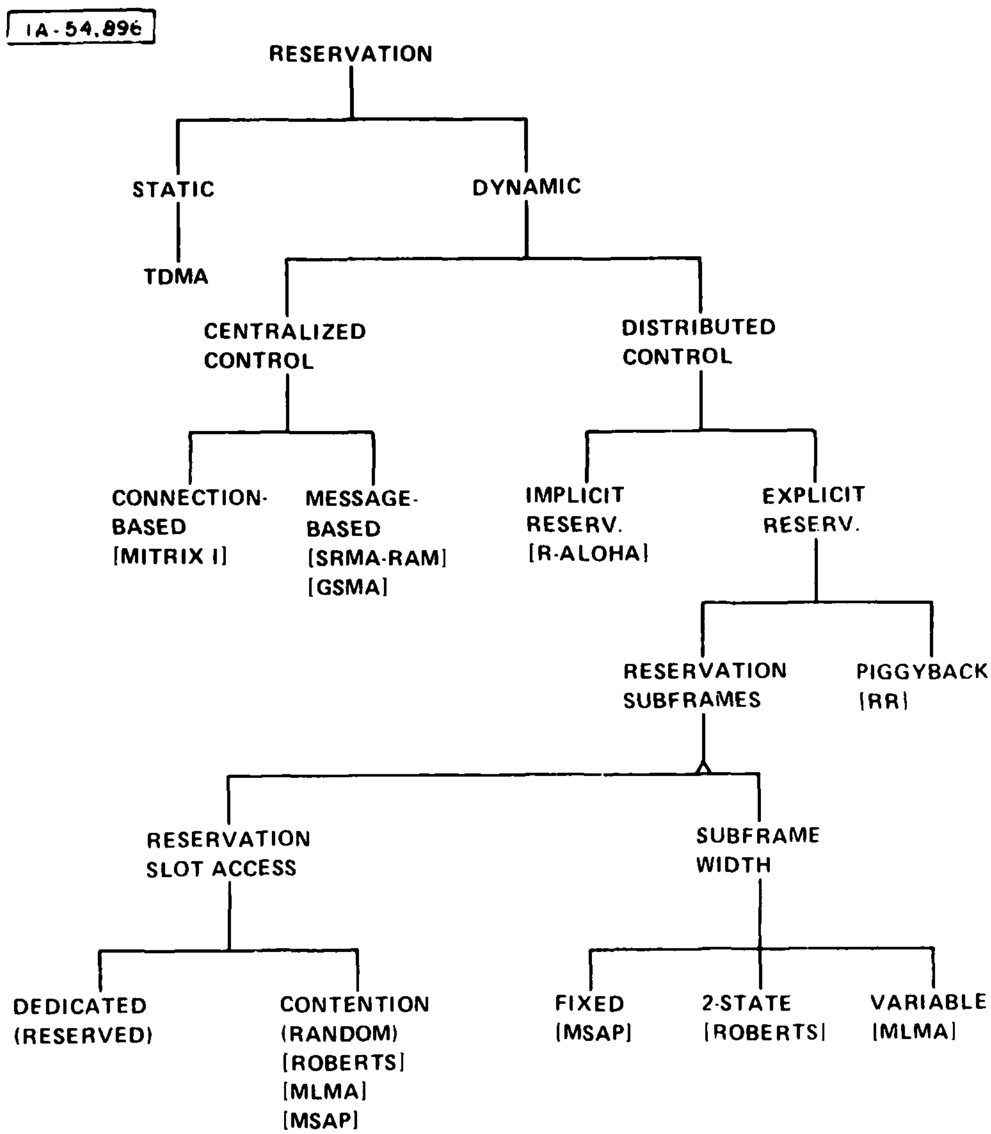
For a more detailed account of these protocols (as well as other bus techniques) consult Luczak's paper [LUCZ78].

Reservation

The third major technique for controlling channel access is known as reservation. In reservation techniques, a node transmits a message (or packet) in a slot that has been reserved for its use. In most reservation techniques, time is "slotted," giving rise to a packet-switching environment. Figure 18 shows Luczak's summary of the various reservation techniques according to their characteristics. Examples of systems employing these access methods are shown. As indicated in the figure, the major distinction between reservation techniques is whether they are static or dynamic.

Time-division multiple access (TDMA) is a well-known static approach. In TDMA, each node is assigned a fixed number of slots per frame. The slots may be assigned to each node according to its requirements. If the nodal requirements are known in advance, this results in high channel utilization. On the other hand, if the data rates are bursty, channel utilization decreases. Digitized audio and video signals are applications which benefit from such a static approach.

In dynamic control, slots are assigned on a demand basis. The two fundamental divisions here are centralized and distributed control. Under centralized control, the nodes make their requests of a central controller, which in turn determines the appropriate number of slots for each node. Distributed control reservation schemes have largely been proposed for satellite systems, as the large propagation times involved in such systems would force a centralized controller to make its decisions on old information.



Based on LUCZ78, p. 71.

Figure 18. Reservation Techniques

The two types of centralized control noted in figure 18 are connection-based control, in which a node requests transmission capability over a period of time, and message-based control, in which the node makes a reservation for each message. The original MITRE bus system, MITRIX I, used a connection-based control system.

Distributed control techniques may be divided into explicit and implicit techniques, depending on whether a special message is or is not required to request slots. Explicit reservations are made via minislots preceding the message slots (reservation subframes) or are made within the message slot (piggyback). When reservation subframes are employed, one must decide which access technique to employ, as well as the width of the subframe.

REVIEW OF CLASSIC MODELS

Here we present a brief summary of the "classic" modeling work done on bus systems (up to 1976). A more detailed presentation of this work appears in Leonard Kleinrock's Queueing Systems (Volume 2) [KLEI76].

ALOHA

The original work on the modeling of broadcast communications systems had its impetus in developing access schemes for satellite communications channels. The ALOHA system at the University of Hawaii inspired the first random access protocol [ABRA73]. Under the ALOHA protocol, if a packet requires P seconds to be transmitted, it will be vulnerable during a period of 2P seconds, as shown by figure 19.

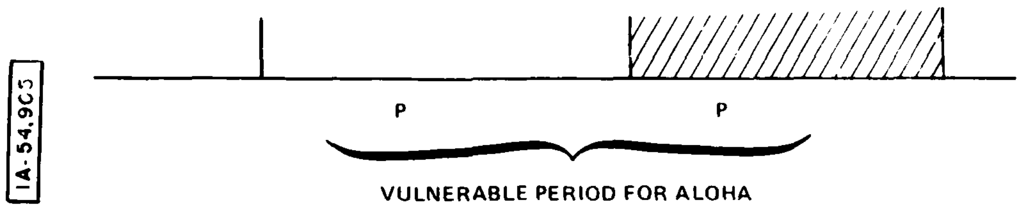


Figure 19. Vulnerable Period for ALOHA

Our model of the ALOHA system will be an infinite population model. Total offered traffic is G packets per transmission period P, where it is understood that each user contributes an infinitesimal amount to the total traffic G. Letting S denote the throughput (number of successful transmissions per P sec), Roberts demonstrated [ROBE73] that S and G are related by the following equation:

$$S = Ge^{-2G} \quad ,$$

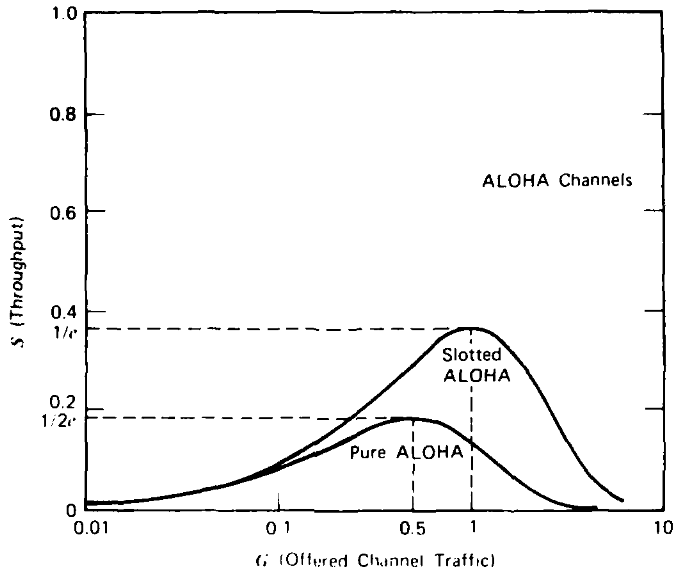
where e^{-2G} is the probability of successfully transmitting a packet. We may see this as follows:

> The probability of successfully transmitting one packet = the probability that no other packets arrive during the vulnerable period 2P. Assuming Poisson traffic, this probability = e^{-2G}.

If we constrain packets to be transmitted only every P sec, then we halve the vulnerable period (P sec instead of 2P sec), and obtain corresponding improvement in the throughput, i.e.:

$$S = Ge^{-G} \quad .$$

Figure 20 demonstrates this relationship.



From KLEI76, p. 365. Reprinted by permission.

Figure 20. Throughput for Pure and Slotted ALOHA

Note that the peak throughput for pure ALOHA is 1/2e (0.18), while for slotted ALOHA we have a peak throughput of 1/e (0.36).

S. Lam has developed a time delay versus throughput model for slotted ALOHA. His work originally focused on satellite channels, but is equally applicable to local broadcast networks [LAMS74].

As in the preceding throughput analysis, we assume an infinite population model. We define the average time delay T to be the average time (in slots) until a packet is successfully received. In our model, a user will broadcast a packet onto the network. If his packet is destroyed, he will rebroadcast the packet randomly during one of the next K slots (with probability 1/K for each slot). This staggering of the packet transmission avoids the situation in which two users collide in a slot and then immediately rebroadcast their packets, thereby ensuring a second (and, in fact an infinite number of) collisions.

It is clear then that the effect of the number of retransmission slots K must be incorporated into a formula for the average packet time-delay. Lam's equation for the average time delay T is therefore

$$T = 1 + E\left(\frac{K+1}{2}\right) \quad ,$$

where E is the average number of retransmission attempts per packet.

Lam then developed an expression for E in terms of the system parameters K and G as well as the throughput S. This expression is:

$$E = \frac{1-q}{q_t} \quad ,$$

where

$$q = \left(e^{-G/K} + \frac{G}{K}e^{-G}\right)^K e^{-S} \quad ,$$

and

$$q_t = \left(\frac{e^{-G/K} - e^{-G}}{1 - e^{-G}}\right)\left(e^{-G/K} + \frac{G}{K}e^{-G}\right)^{K-1} e^{-S} \quad .$$

Lam also developed an expression for the throughput S in terms of q, q_t, and G, given by

$$S = G\,\frac{q_t}{1 + q_t - q} \quad .$$

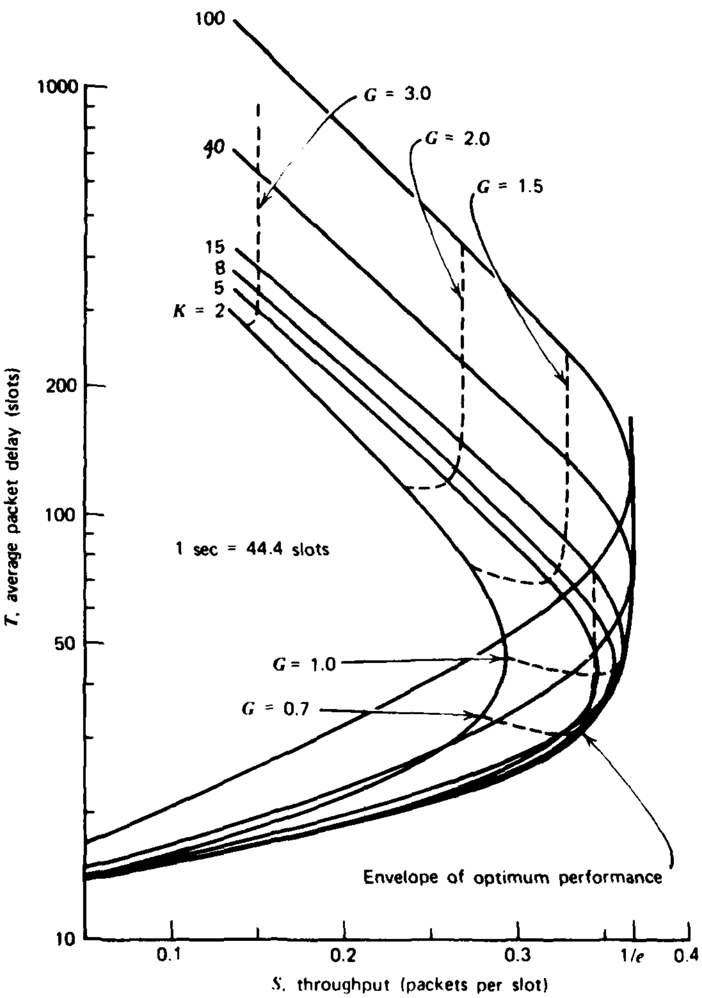
For a derivation of these equations, the reader may wish to consult LAMS74 or SCHW77.

Ideally one would wish to solve the equations for q, q_t, and S simultaneously in terms of the system parameters G and K, thereby obtaining an expression for the time delay T in terms of these parameters. This is a difficult task, hence one must settle for numerical results.

The fundamental relationships among the four parameters T, S, K, and G are depicted in figure 21. The dotted lines correspond to constant G (offered load) contours. We note that the effect of increasing the number of packet retransmission slots K is to increase throughput.

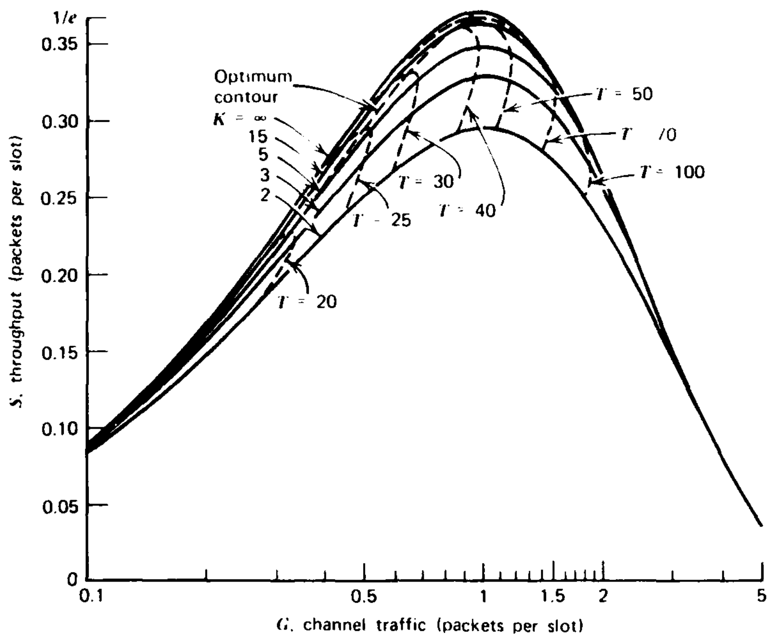
Once K is increased past 15, the increase in throughput is infinitesimal, while the time delay increases precipitously. Fixing K at any value, we note that the maximum throughput occurs at G = 1.0. Attempting to increase the channel load beyond this point rapidly drives up the time delay.

Figure 22 shows an alternate way of viewing this trade-off. This diagram is interesting because of its constant time-delay contours. One can increase the throughput while maintaining a constant time delay by increasing K -- provided, again, one does not increase K too far. An optimal contour relating these three parameters is displayed in the figure.



From KLEI76, p. 374, Reprinted by permission.

Figure 21. Delay-Throughput Trade-off



From KLEI76, p. 373. Reprinted by permission.

Figure 22. Throughput as a Function of Channel Traffic

Stability Considerations

As may be observed from figure 21 (which portrays the equilibrium time-delay versus throughput trade-off), one can find two possible time delays for given values of S and K. The smaller of the two is called the channel operating point, because it represents the ideal point at which to operate the channel in light of the time delay versus throughput results. The existence of two time delays for a single throughput suggests that the channel equilibrium assumption is not valid. To investigate this possibility, Kleinrock and Lam simulated an infinite population model. The simulations revealed the following behavior: Beginning with an empty system, the channel proceeds into equilibrium at the channel operating point for a finite time period, after which stochastic variations in the traffic arrival pattern increase the traffic load, resulting in decreased throughput and higher packet delays. This pattern repeats itself until the channel eventually drifts into saturation, with the throughput going to zero.

These observations prompted the realization that the fundamental trade-off for slotted ALOHA is not time delay versus throughput, but instead, <u>time delay versus throughput versus stability</u> [LAMS74 or KLEI75].

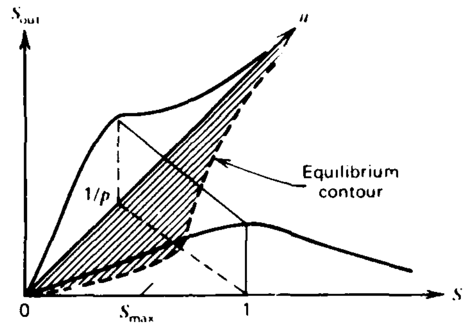
In developing a model for this trade-off, Kleinrock and Lam developed a linear feedback model. The channel state for this model is portrayed as dependent on two random variables:

1. N(t), representing the total number of busy terminals at time t.

2. S(t), the combined input packet rate at time t.

If $N(t) = n$, then $S(t) = (M - n)\sigma$ where M is the number of active terminals (having a packet to transmit) and σ is the probability that a given terminal will transmit a packet in a slot.

Using this model, Kleinrock and Lam developed an equation representing the throughput in terms of the parameters N and S. Figure 23 illustrates this relationship. It is important to note that the throughput S_o represented in this figure is <u>not</u> the same as the equilibrium throughput S. In this context S is the amount of new channel traffic, i.e., the input. The shaded region represents a safe region, in which the throughput (S_o) exceeds the input, whereas the unshaded region represents the situation in which the system's capacity is exceeded.



From KLEI76, p. 378. Reprinted by permission.

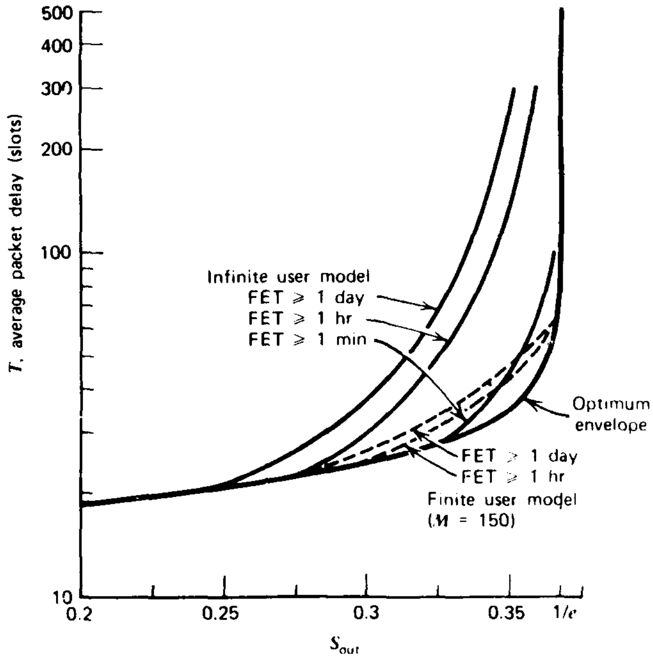
Figure 23. Channel Throughput Rate as a Function of Load and Backlog

With this in mind, two categories of channels can be defined:

1. Stable channels, in which equilibrium results always obtain. A finite population of users falls into this category, provided that a sufficiently high value for K is chosen. Choosing higher values for K results, of course, in higher time delays.

2. Unstable channels, in which the equilibrium conditions hold only for a finite period of time, after which the channel drifts into an overload situation. As already indicated, the infinite population model is unstable.

To provide a numerical measure of instability, Lam defined the first exit time (FET) as the time required to exit into the unsafe region, starting from a zero backlog. With this definition in mind, it is possible to represent the fundamental time delay versus throughput versus stability trade-off by figure 24.



From KLEI76, p. 383. Reprinted by permission.

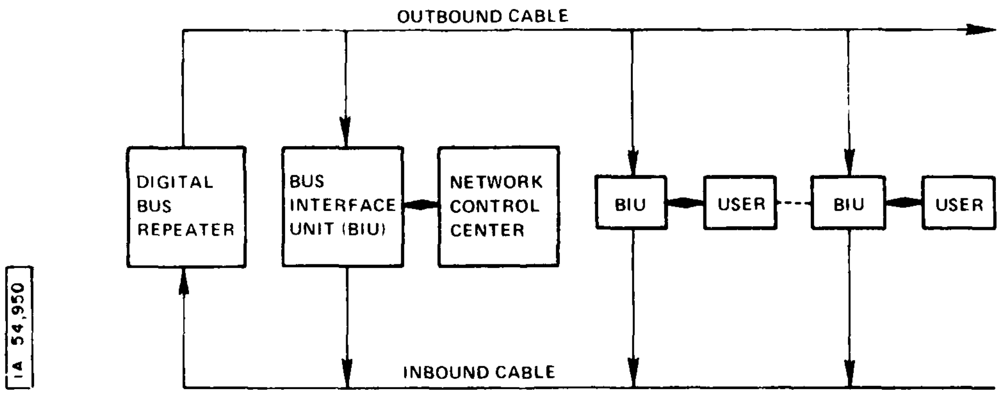
Figure 24. Stability-Throughput-Delay Trade-off

The lower solid line represents the optimal envelope obtained from the time delay versus throughput trade-off. The three solid curves represent an infinite population model corresponding to three different FET values of ≥ 1 day, ≥ 1 hour, and ≥ 1 minute, while the two dotted lines represent a finite population model (M = 150) with FET values of ≥ 1 day and ≥ 1 hour.

As is indicated by these curves, a better time delay versus throughput performance is achieved at the expense of more frequent channel overloads. If occasional channel overloads are acceptable, then the channel need only be shut down and restarted when such a situation occurs. If this solution is not acceptable, Lam has developed dynamic control procedures to insure system stability. These are reported in LAMS74.

MITRE Cable Bus System

The slotted ALOHA protocol has been implemented on a cable bus system (referred to as MITRIX2) developed at MITRE. Each user in the system is attached via an interface unit to two unidirectional cables. One of the cables is employed to carry data from the users to a repeater located at the head-end of the system. An outgoing cable carries the messages from the repeater back to the users. The transmission rate is 7.373 Mbit/sec. Figure 25 is a diagram of this system.



Based on MEIS77, p. 5-14.

Figure 25. MITRIX Bus Structure

MITRIX2 is a dual-mode system, built to accommodate both high and low duty-cycle users. Synchronous, high duty-cycle users have dedicated slots assigned to them while the bursty, low duty-cycle users are assigned a set of slots which they compete for in ALOHA fashion. Control procedures based upon Lam and Kleinrock's work are employed on the ALOHA subchannel. A description of the system may be found in MEIS77a.

Modeling of CSMA Protocols

As mentioned previously, with CSMA protocols, the users listen to the channel and wait until it is idle before broadcasting their messages onto the network. The impetus for a protocol of this nature lies in the fact that the propagation delay in a local network is much smaller than the transmission delay, and one may therefore obtain up-to-date information about the state of the channel. For example, if a 1000-bit packet travels over a 10-mile, 100 Kbps channel, one obtains a transmission time of 10 msec and a maximum propagation delay of 0.054 msec. This situation is a great contrast to a satellite communications channel, in which the propagation delays are of the order of 0.25 sec. Listening to a satellite channel provides information about the past. In fact, it provides such ancient history that attempting to employ a CSMA protocol on such a channel would be deleterious, since decisions would be made on out-of-date information.

An extension of the CSMA protocols has been developed in which the users continue to monitor the transmission line during their message transmission and immediately cease transmission upon detecting a collision. This protocol is called the Listen-While-Talk (LWT) protocol.

As the two major categories of persistent and nonpersistent CSMA protocols have already been discussed in some detail, we will simply restate the algorithms for nonpersistent, 1-persistent, and p-persistent CSMA in order to refresh the reader's memory.

In a nonpersistent protocol, a ready node (one having a message to transmit) first senses the channel. Then:

1. If the channel is sensed idle, it transmits its message.

2. If the channel is sensed busy, it reschedules the message for some later time, picking a delay time from a retransmission distribution. It then repeats step one.

In a 1-persistent protocol, a node with a message to transmit continues to sense a busy channel until the channel becomes idle. Then the node transmits a message.

In a p-persistent protocol, time is slotted into slots of length equal to the maximum propagation delay. A ready node senses the channel, and:

1. If it is idle, the node broadcasts the packet with probability p, and delays one slot with probability (1 - p), at which point it again senses the channel. If busy, it continues to sense the channel, and repeats this step.

2. If the node is delayed one slot, it then repeats step 1.

Our first category of models is aimed at developing a throughput (denoted by S) versus offered load (denoted by G) trade-off. In doing so, we develop throughput equations for S in terms of G and other system parameters. These equations were developed by F. Tobagi in his thesis [TOBA74].

In developing these equations, Tobagi assumes an infinite population model generating an average of G packets per P sec. Other assumptions are:

- Each packet is of constant length.

- Each terminal has <u>exactly</u> one packet awaiting transmission.

- The acknowledgment channel is separate from the data channel.

- All source-destination pairs are assumed to have the same one-way delay. This delay is normalized with respect to transmission time on the channel.

A complete list of the model's assumptions may be found in TOBA74.

The propagation delay is denoted by a, and is equal to $\frac{d/2}{P}$, where d/2 is the one-way propagation delay and P is the packet transmission time.

We develop an equation for S in terms of G for nonpersistent CSMA to give an indication of the modeling approach employed in developing the throughput equations for the CSMA protocols. The interested reader may consult Tobagi's thesis for the remaining equations as well as their derivations. In LABA78, LaBarre points out an error in Tobagi's throughput equation for nonpersistent CSMA and corrects it. Hence we present LaBarre's derivation of this equation.

In analyzing nonpersistent CSMA, LaBarre notes that the activity on the channel may be divided into busy and idle periods. The combination of one busy and one idle period is the channel cycle.

The following equation from renewal theory enables LaBarre to calculate the throughput:

$$S = \frac{\bar{U}}{\bar{B} + \bar{I}} ,$$

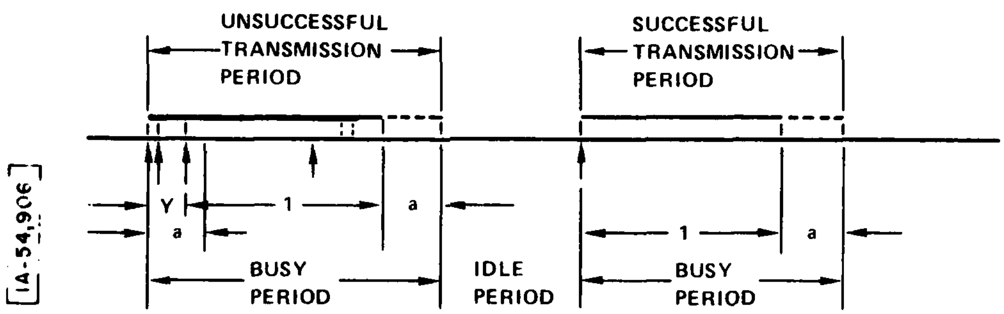
where

$\bar{U}$ = average time during the cycle in which the channel is used successfully.

$\bar{B}$ = average length of the busy period.

$\bar{I}$ = average length of the idle period.

To calculate these quantities, we refer to figure 26 [TOBA74]. In this figure, a packet is pictured arriving at time 0. It is subject to a number of collisions during the first busy period, resulting in an unsuccessful transmission period. The last arrival occupies the channel for 1 (normalized) transmission delay, and clears the channel a (normalized) time units later.



Based on TOBA74, p. 55.

Figure 26. Nonpersistent CSMA: Busy and Idle Periods

The busy period is followed by an idle period (mean length 1/G) and is immediately followed by another busy period. In this busy period, the transmission is successful, hence the length of this period is 1 + a.

We start with $\overline{U}$. The fraction of time during a cycle in which the channel is used without interference is equal to the probability that no packets arrive during the first seconds of the packets' transmission. Since we assume an average arrival rate of G packets per P sec, the Poisson distribution yields e^{-aG}. (Recall that $a = \frac{d}{2P}$.)

To calculate $\overline{I}$, we note that the interarrival time must be 1/G (Poisson process).

To calculate $\overline{B}$, we first define the random variable Y in the interval (o, a) to be the instant at which the last packet that collides with our first arrival appears. If there are no collisions with our first arrival (i.e., a successful transmission), then the length of the busy period is 1 + a. The average length of the unsuccessful busy period is equal to $1 + \overline{Y} + a$, where $\overline{Y}$ is the mean value of Y. We compute $\overline{Y}$ as follows:

The distribution function $F_Y(y)$ is given as

$$F_Y(y) = P \text{ (no arrivals occur in an interval of length } a - y).$$

$$= e^{-(a-y)G} \quad .$$

The mean of this distribution, $\overline{Y}$, is given by

$$\overline{Y} = a - \frac{1}{G}\left(1 - e^{-aG}\right) .$$

$\overline{B}$ may now be calculated as follows:

$$\begin{aligned}\overline{B} &= P \text{ (successful transmission) } (1 + a) \\ &\quad + P \text{ (unsuccessful transmission)} \\ &\quad \cdot (1 + \overline{Y} + a) \quad , \\ &= e^{-aG}(1 + a) + \left(1 - e^{-aG}\right)(1 + \overline{Y} + a) \quad , \\ &= 1 + \left(1 - e^{-aG}\right)\overline{Y} + a \quad .\end{aligned}$$

Substituting values for $\overline{I}$, $\overline{B}$ and $\overline{U}$ in our throughput equation, we obtain the following expression for S:

$$S = \frac{Ge^{-aG}}{G\left(1 + 2a - ae^{-aG}\right) - \left(1 - e^{-aG}\right)^2 + 1} .$$

Tobagi obtains throughput equations for the other CSMA protocols he defines in his thesis (1-persistent and p-persistent CSMA), as well as slotted versions of these protocols. The interested reader should consult Tobagi's thesis [TOBA74] or Kleinrock and Tobagi's article, "Packet Switching in Radio Channels" [KLEI75b].

LaBarre [LABA78] develops a throughput equation for the nonpersistent Listen-While-Talk (LWT) CSMA protocol. His equation is:

$$S = \frac{Ge^{-aG}}{G\left(e^{-aG} + a\right) + (1 + aG)\left(1 - e^{-aG}\right)^2 + 1} .$$

Figure 27 shows LaBarre's graph comparing the nonpersistent LWT protocol with the other CSMA protocols.

LaBarre points out that the nonpersistent LWT protocol offers a 10 to 30 percent improvement in maximum throughput for propagation delays a ranging between 0.01 and 0.05. Tobagi presents a similar diagram (figure 28) which includes p-persistent CSMA as well as slotted nonpersistent CSMA.

As can be seen by comparing figures 27 and 28, nonpersistent LWT has the edge in throughput performance over the p-persistent CSMA protocols also.

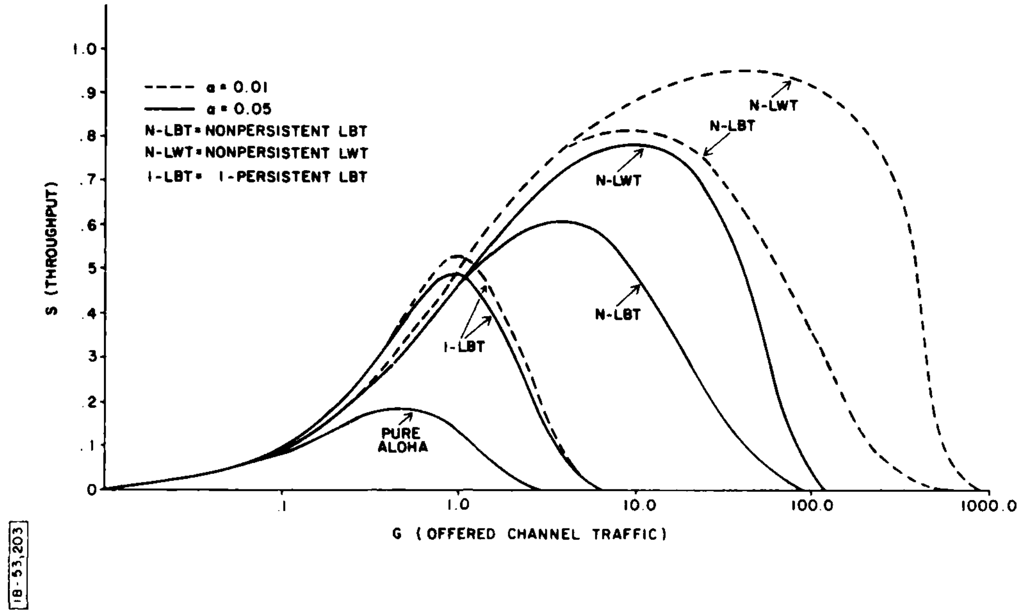
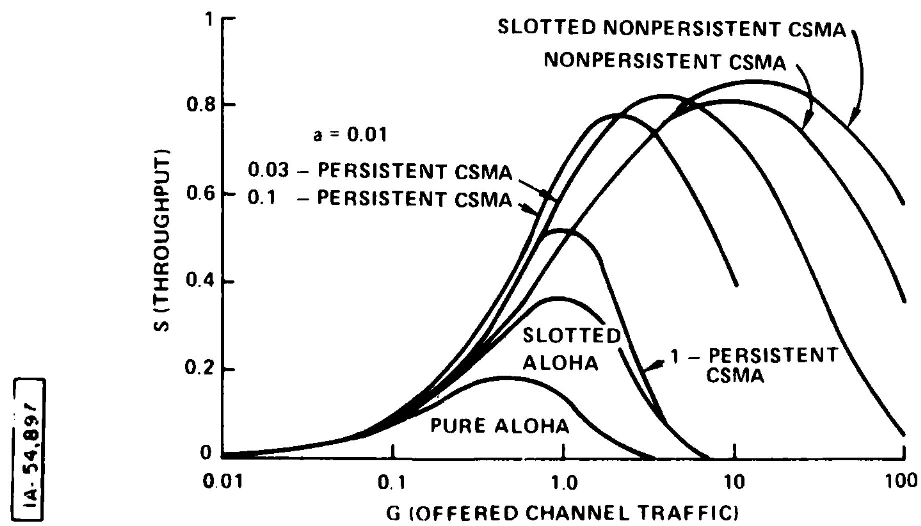


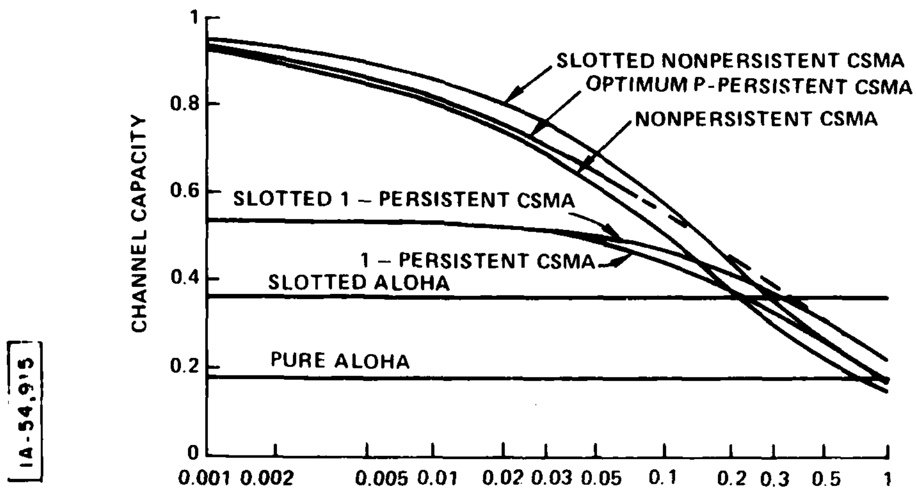
Figure 27. Throughput Versus Offered Load -- Analytic Results



Based on KLE176, p. 400 Used by permission.

Figure 28. Throughput Versus Channel Traffic

It is important to note the effect of parameter a on the throughput of these protocols. Increasing the value of a does not affect the ALOHA protocols but it does significantly affect the CSMA protocols. The reason (pictured in figure 29) is that the large values of a correspond to older information. Hence, decisions on whether or not to broadcast a packet will be made on incorrect information. (Never trust second-hand information -- it degrades performance.)



Based on KLEI76, p. 400. Used by permission.

Figure 29. Effect of Propagation Delay on Channel Capacity

We now turn to time delay versus throughput models. Making two additional assumptions (discussed below), Tobagi develops equations for the time delay in terms of the system throughput, the offered load and other system parameters. We present his equation for non-persistent CSMA, and refer the reader to TOBA74 or TOBA75 for the equations for the remaining CSMA protocols, as well as their derivations.

In order to derive the equation, Tobagi assumes:

1. The average retransmission delay $\overline{X}$ is large when compared to the transmission time T.

2. The interarrival times of the point process (defined by the start times of all the packets plus transmissions) are independent and exponentially distributed.

3. The acknowledgment packets are correctly received with probability 1. (One might create a separate channel for acknowledgments.)

4. The processing time required to perform the sumcheck and to generate the acknowledgment packet is negligible.

Let W be the (normalized) time delay for an acknowledgment packet, and let ξ be the normalized mean retransmission delay. Then the expected time delay D must be the sum of:

1. The transmission time, given by $1 + a$.

2. The expected delay due to deferring (i.e., sensing a busy channel). The expression for this mean delay is $\frac{G - H}{S}\xi$, where H is the amount of traffic the channel attempts to transmit.

3. The expected retransmission time, given by $[(H - S)/S](1 + 2a + W + \xi)$. The second term in parentheses represents the sum of a transmission, an acknowledgment, and a retransmission delay.

Hence, the expression obtained for D is

$$D = 1 + a + \left(\frac{H - S}{S} \right) (1 + 2a + W + \xi) + \left(\frac{G - H}{S} \right) \xi \; .$$

H may be computed as follows:

$$H = G\left(1 - P_b\right) \quad ,$$

where P_b is the probability of being blocked. An expression for $1 - P_b$ is given by

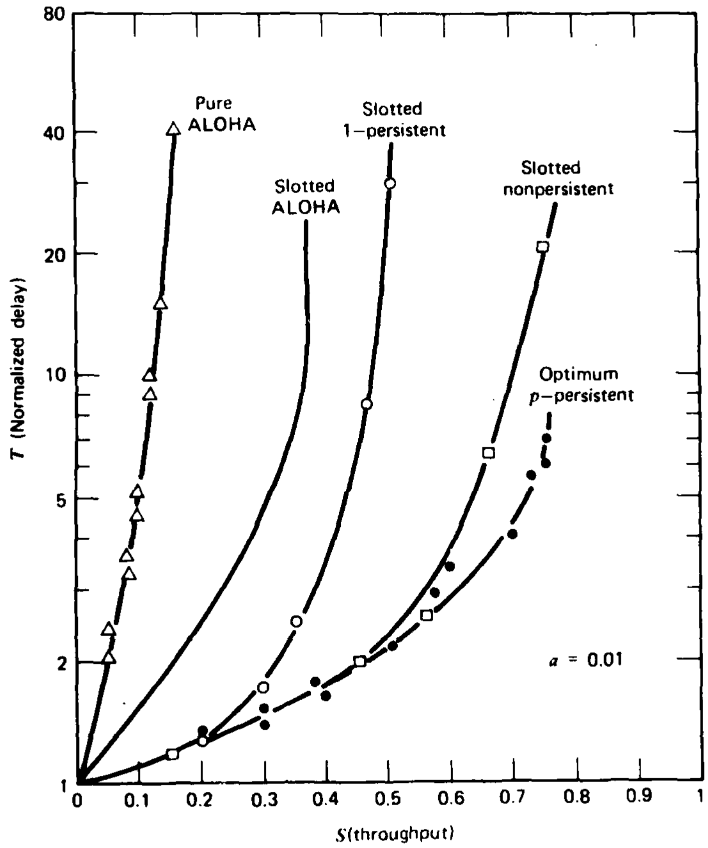
$$\frac{1 + aG}{1 + G(1 + a + \overline{Y})} \quad .$$

It is possible (although difficult) to obtain optimal values of ξ such that D is minimized for a given value of S. This may be seen from the fact that G/S is a decreasing function of ξ.

A simulation was performed which presents the basic time delay versus throughput trade-off but with assumptions 1 and 2 dropped. The result of this simulation is presented in figure 30 (for a = 0.01).

The best performance is obtained from the optimal p-persistent protocol.

LaBarre [LABA78] develops an equation for nonpersistent LWT. He then uses the analytic delay results to perform the same trade-off. However, he replaces the p-persistent protocol with the LWT protocol, and concludes that the LWT protocol is best. His graph is reproduced in figure 31.



From KLEI76, p. 401. Reprinted by permission.

Figure 30. Time Delay Versus Throughput

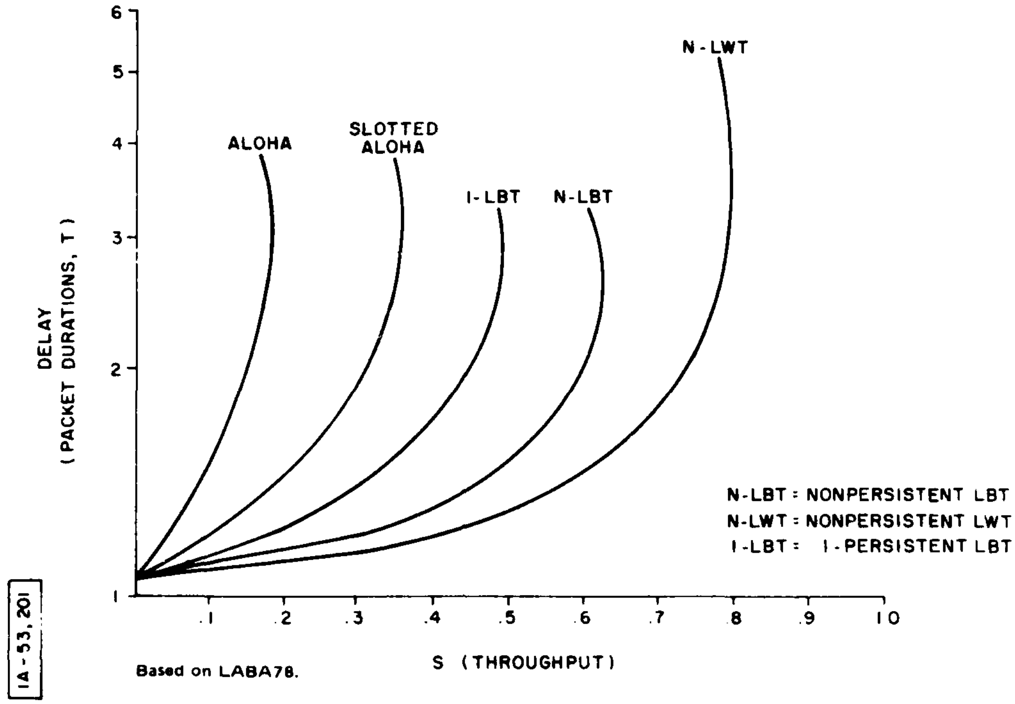


Figure 31. Analytic Delay Results (α = 0.05, $\bar{\xi}$ = 0.12)

The 1-persistent LWT protocol has been implemented on the MITRE bus system.

Simulation studies of the system were performed in which the 1-persistent LBT protocol was compared to the 1-persistent LWT protocol [LABA78]. The results indicated that the LWT protocol has a maximum throughput of more than twice that of the LBT protocol, and a time delay of less than half of the LBT protocol for corresponding throughput values.

Tobagi and LaBarre point out that the dynamic control procedures developed by Lam for use in a slotted ALOHA channel are applicable to the CSMA and LWT protocols [TOBA74 and LABA78].

The LWT protocol has been implemented in three systems:

- Ethernet -- Developed at the Xerox Corporation, Ethernet connects up to 256 communicating computers at 3 Mbits/sec over 1 km of coaxial cable, utilizing off-the-shelf CATV taps and connectors. A description of Ethernet may be found in METC76.

- Fibernet -- Also developed by Xerox, Fibernet makes use of optical fibers to connect up to 19 stations at 150 Mbits/sec through 1/2 km of optical fiber. A description of Fibernet may be found in RAWS78.

- MITRE Cable Bus -- A brief description of this system is given in the section of this paper on the ALOHA protocol.

HYPERCHANNEL -- A CSMA NODAL PRIORITY ACCESS SCHEME

Network Systems Corporation of St. Paul, Minnesota, has designed a prioritized-access local network, called Hyperchannel.

This network was developed to alleviate the bottleneck caused in many computer centers when one large computer was assigned the task of controlling numerous storage devices for the remaining processors on the site.

In Hyperchannel, each port connects through an adapter (which performs functions such as buffering, flow control, etc.) to as many as four coaxial cables, each having a line speed of 50 Mbit/sec. Each coaxial cable can accommodate 16 hosts up to 1000 ft apart. Figure 32 is a diagram of this system, taken from THOR75. For more details, the reader should consult this paper.

AD-A085 282 MITRE CORP BEDFORD MA F/G 17/2
MODELS OF LOCAL COMPUTER NETWORKS.(U)
APR 80 C TROPPER F19628-79-C-0001
UNCLASSIFIED MTR-3783 ESD-TR-80-111 NL

2 OF 2
AD A
085282

END
DATE
FILMED
7-80
DTIC

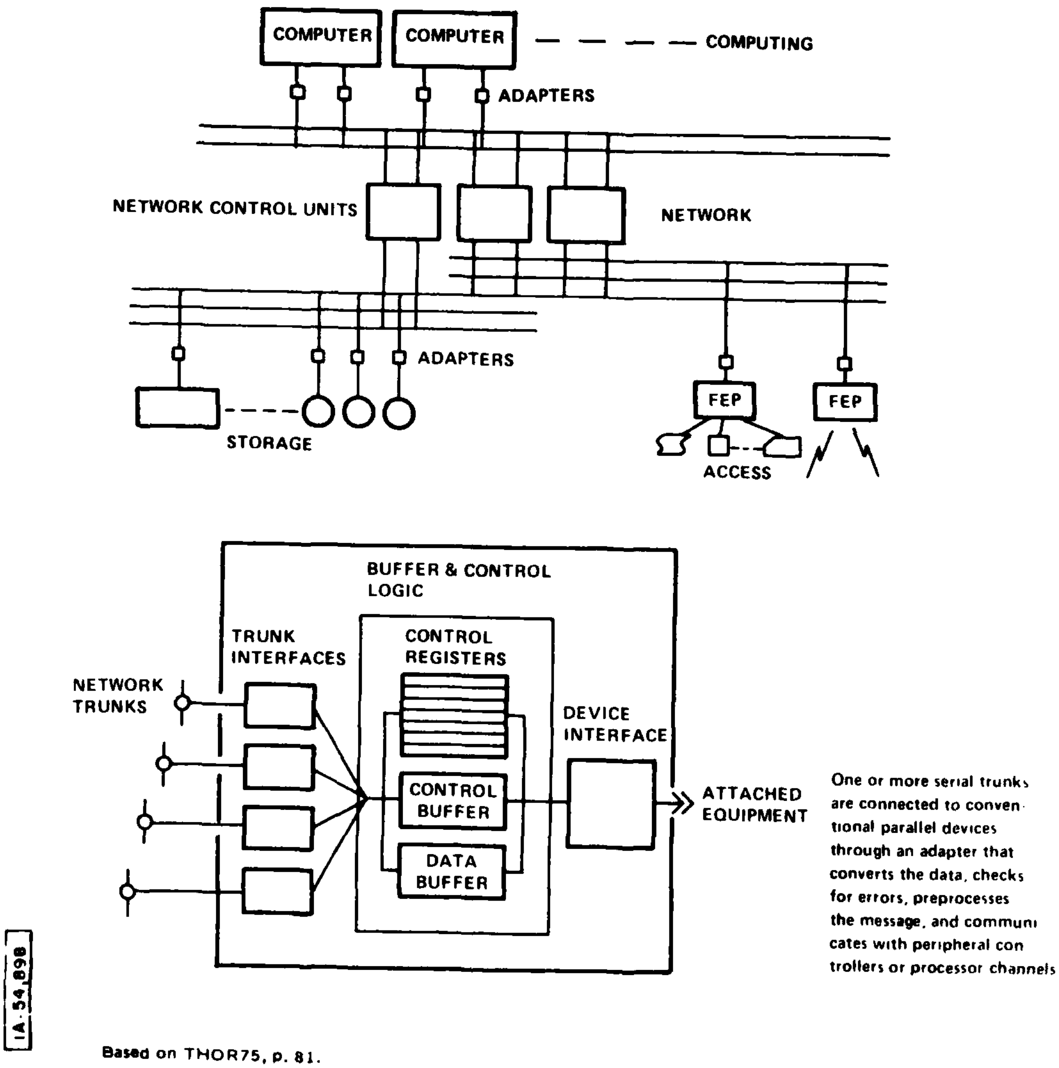


Figure 32. Hyperchannel Architecture

The access protocol employed by Hyperchannel is described below.

For the i^{th} CIU* we have:

```
Loop:    Wait until message to transmit;

Loop 2:  If carrier absent then [transmit message;
                                 if collision then
                                 [wait Di time units;
                                 goto Loop 2] else goto
                                 Loop]
                            else [wait until carrier
                                 absent; wait Di time
                                 units; goto Loop]

         goto Loop:
```

$0 < D_1 < D_2 < \ldots < D_n$

$D_{i+1} - D_i > d$

with d representing maximum length path propagation delay [FRAN77, p. 3].

Extensive modeling of Hyperchannel has been performed. A queueing model was developed by Franta and Bilodeau ("Analysis of a Prioritized CSMA protocol based on Staggered Delays") [FRAN77], and a number of simulation studies have been conducted at the Lawrence Livermore Laboratory of the University of California [DONN78a, DONN78b and NESS78]. We will discuss both the queueing model and the simulation studies and compare their predictions about the system.

Franta and Bilodeau were interested in obtaining an expression for the steady-state time delay suffered by the i^{th} CIU in the system, as well as the steady-state throughput. They based their approach upon

*In this description, the adapter is referred to as a CIU (communications interface unit).

Tobagi's analyses of both persistent and nonpersistent CSMA protocols [TOBA75]. This approach is based upon renewal theory [COX62]. A cycle on the channel is pictured as consisting of alternating idle and busy periods.

The steady-state throughput is given by

$$S = \sum_{i} P[X = i] \frac{U(i)}{I(i) + B(i)} \quad .$$

In this formula:

- i is the state of the system. This state is defined as a vector with N components where N = the number of CIUs sharing the channel. The state of each of the N CIUs sharing the channel may be represented by a 0, 1, or a 2 -- indicating that the CIU is idle, that it is blocked,* or that it has started the current busy period.

- U(i) is the period of time during a cycle that the channel is conflict-free.

- B(i) is the length of the busy period.

- I(i) is the length of the idle period.

Each of these steady-state expressions is derived by obtaining an expression for the appropriate behaviors, and then passing to the limit.

*That is, has a message to transmit, but cannot do so because the channel is busy. A 1 is also used to denote that a transmission was initiated by the CIU, but that it will be unsuccessful.

To obtain the transient expressions for the n^{th} cycle, one must evaluate two categories of expressions:

- transition probabilities of the form

$$P\left(X_{n,k} = j/X_{n,k-i} = i\right) \quad ,$$

  where $X_{n,k}$ represents the state of the system at the k^{th} stage of the n^{th} cycle.

- time delays that occur between stages of the cycle.

The interested reader should consult FRAN77 for the derivations of the quantities.

In order to derive the steady-state probabilities $P(X = i)$ that are used in the throughput equation, Franta and Bilodeau solve the classic equation for steady-state probabilities in a discrete-time Markov chain, i.e.,

$$\pi P = \pi \quad .$$

Note that this involves solving 2^N equations in 2^N unknowns (N = number of CIUs) -- the authors have chosen the brute-force approach.

Next the authors obtain an expression for the waiting time W_i of the i^{th} CIU, defined to be the interval between the time when the CIU first makes the transition from the idle state to the busy state, and the time when it makes the reverse transition.

W_i is calculated via the following formula

$$W_i = \sum_{j=o}^{\infty} Q_i(j) \cdot R_i(j) \quad ,$$

where

j = the j^{th} busy period,

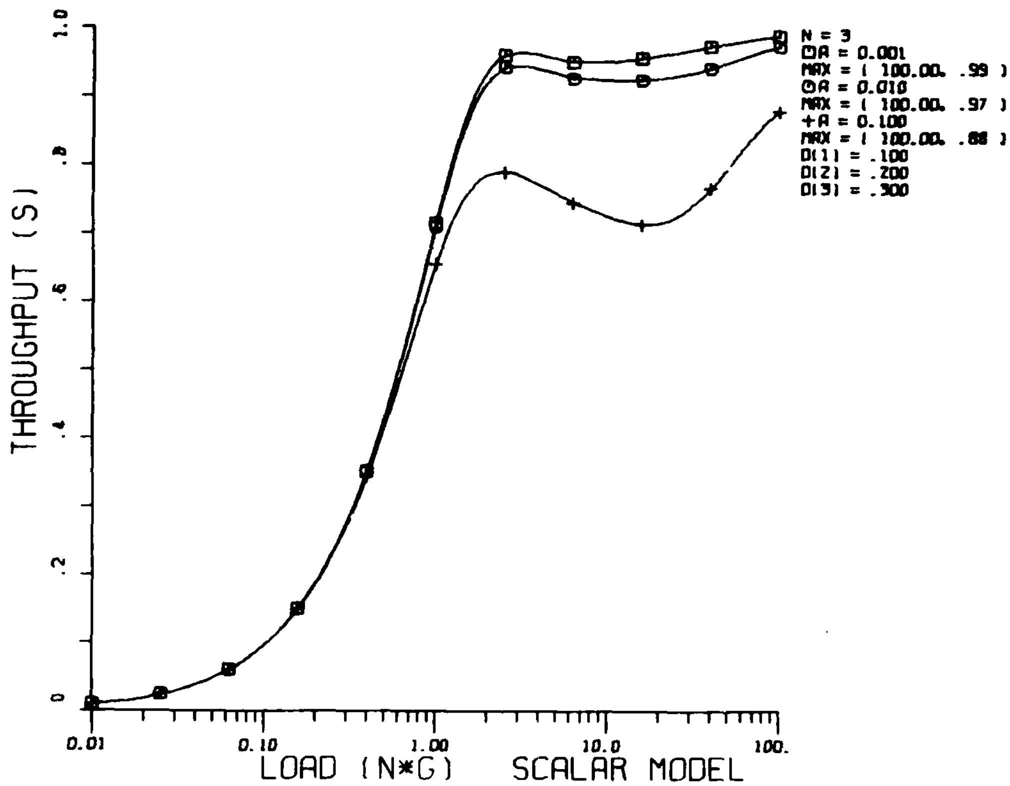
$Q_i(j)$ = probability that j successfully transmits in the j^{th} busy period following becoming busy,

$R_i(j)$ = length of time between becoming busy and becoming idle.

After solving the above collection of equations, Franta and Bilodeau present a number of conclusions concerning the behavior of Hyperchannel. We begin with their discussion of throughput versus offered load. Figure 33 is based on an illustration from their paper.

As shown in figure 33, the relationship between throughput and offered load may be divided into three regions: In the first region, the throughput increases with increasing load. The idle period on the channel decreases with increasing load, thus permitting the throughput to grow. (The utilization and the busy period remain relatively constant.)

In region two, the idle period can no longer be decreased to accommodate increasing traffic, as the channel's entire bandwidth has been used. An increased number of collisions results, and throughput drops.



From FRAN77, p. 44.

Figure 33. Throughput Versus Offered Load

Region three contains the apparent paradox of increased throughput at still higher loads. The explanation for this, however, lies in the Hyperchannel's priority access protocol. With increased loads, the CIU with the highest priority dominates the channel to the exclusion of the remaining CIUs.

With respect to time delay/throughput trade-offs, Hyperchannel behaves in very much the same fashion as other random access protocols (slotted ALOHA, CSMA). That is, in order to efficiently operate Hyperchannel, one must search for the "knee" of the time delay/throughput curve. The highest throughputs once again correspond to the highest time delays. Consult KLEI76 for a discussion of the throughput/time delay/stability trade-offs in random access protocols. A summary of these trade-offs is included earlier in this section.

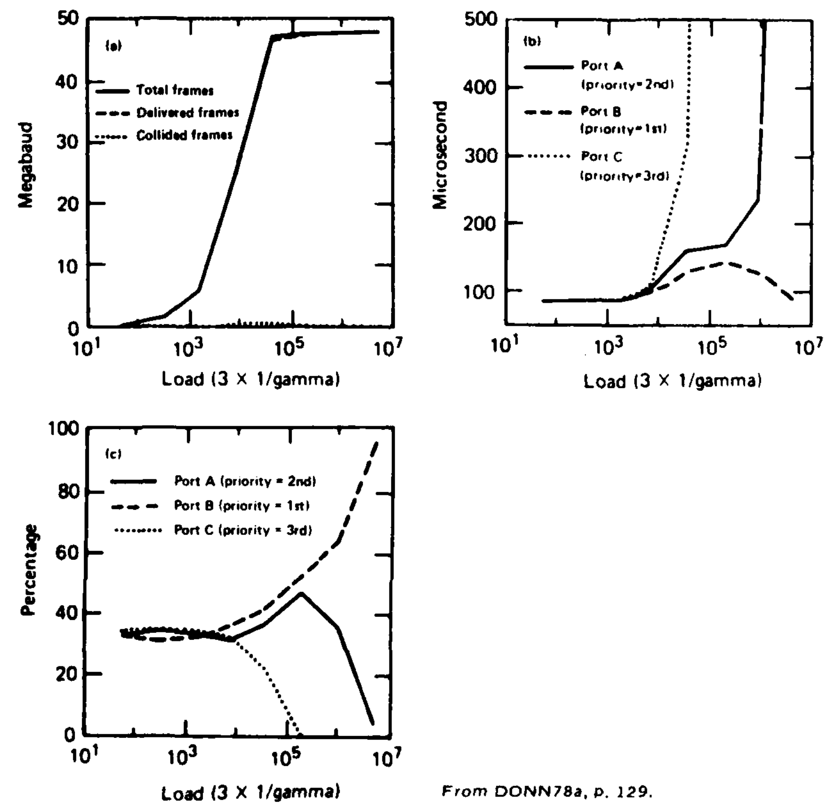
Simulation studies conducted by Donnelley and Yeh at Lawrence Livermore Laboratories [DONN78a] are in close agreement with Franta and Bilodeau's results. Figures 34 and 35 reproduce several performance curves from these studies. The simulations were written in ASPOL, and run on a CDC 7600 computer.

More detailed simulations were conducted by Donnelley and Yeh on the higher level protocols governing flow control, recovery from errors, etc. (termed by the authors the level 2 protocols). This work was continued in DONN78b. In both papers, the authors point out problems (and suggested solutions) in the level 2 protocol <u>which tended to negate the good performance of the level 1 protocol</u>. Their major conclusion is that it is vital to construct the level 2 protocols in a prioritized CSMA system such as Hyperchannel in such a way as to complement the level 1 (trunk access) protocol [DONN78b]. In NESS78, a second-level protocol is introduced to alleviate some of the problems uncovered in DONN78a and DONN78b.

GAMMA (sec)	Rate=1/GAMMA	Load=Adapters•Rate
50,000E-6	20	6E+1
10,000E-6	100	3E+2
2,000E-6	500	1.5E+3
400E-6	2,500	7.5E+3
80E-6	12,500	3.75E+4
16E-6	62,500	1,875E+5
3.2E-6	312,500	9.375E+5
0.64E-6	1,562,500	4.6875E+6

From DONN78a, p. 129.

Figure 34. Traffic Load



From DONN78a, p. 129.

Figure 35. Performance Curves (semi-log plot) for 3-Port Network (message length = 4K bits): (a) Trunk Throughput, (b) Frame Delay of Each Port, and (c) Throughput Percentage of Each Port

CONFLICT-FREE RESERVATION SCHEMES

All the access schemes discussed so far have been characterized by messages subject to collision during a certain period of time, with a resulting loss in throughput. Some schemes have recently been introduced in an attempt to eliminate collisions, and thereby improve system performance. The salient feature of these schemes is that they employ some form of message reservation scheme. Time is divided into frames, which are in turn subdivided into reservation slots and message slots.

As their name indicates, the reservation slots are employed by each user to reserve space for the packets it must send. (We assume that messages are divided into packets.) The message slots are used to accommodate the user-generated packets. Control of these systems may be either centralized or distributed. In a centralized environment, a scheduler allocates message slots on the basis of the requests, while in a distributed system, some form of nodal prioritization is employed.

Other reservation schemes were proposed earlier than those discussed in this section. Roberts proposed a centralized system for satellite packet-switching, in which reservation slots were contended for in ALOHA fashion [ROBE73]. A centralized FDMA reservation scheme referred to as split-channel multiple access (SRMA) was proposed by Tobagi for packet radio systems [TOBA74]. In SRMA, two channels are created: a contention channel for message reservation and a TDMA channel for message transfer. Both Roberts' and Tobagi's schemes employ random access methods for message reservations, while the schemes discussed in this paper employ other techniques to resolve contention for message reservation.

We discuss four schemes in this section. The first, the mini-slotted alternating priorities (MSAP) scheme, was developed by Kleinrock and Scholl at UCLA. It is used in a decentralized environment and is especially effective for a small number of buffered users (≤ 50). A discussion of a family of protocols closely related to MSAP is also presented, as they provided the motivation for MSAP's development.

The MLMA (Multi-Level Multi-Access) protocol, developed at IBM (Zurich) is discussed next. This will be seen to be related to MSAP's precursors. MSAP is intended for use in a distributed environment.

The third protocol to be discussed will be the GSMA (Global Scheduling Multiple Access) protocol, developed at IBM's Watson Laboratories. GSMA employs a centralized scheduling mechanism.

The last protocol discussed is the DYN (Dynamic Reservation) protocol, suggested by Kleinrock. DYN is intended for use in a distributed environment, and is shown to be particularly effective in a heavy traffic environment.

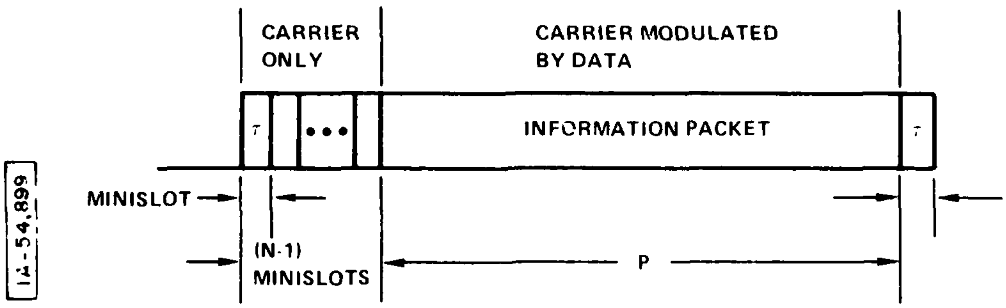
MSAP

Our description of MSAP (mini-slotted alternating priorities) is based upon a paper by Kleinrock and Scholl [KLEI77a].

MSAP grew out of an attempt to provide good delay performance for a small number of users (~ 20) at higher traffic loads. The CSMA protocols tend to provide good delay performance at low loads, but performance degrades at higher loads as a result of the increasing number of collisions. At the other extreme, one finds the fixed assignment systems of time-division multiple access (TDMA) and frequency division multiple access (FDMA), which are suitable for high traffic levels but perform poorly at lower traffic levels.

A family of three protocols was developed that solved the above problem. Known as the Alternating Priorities (AP), Round Robin (RR), and Random Order (RO) protocols, they all use carrier sensing in the following manner.

Time is divided into frames, as indicated in figure 36.



Based on KLEI77, p. 22.1-106.

Figure 36. Slot Configuration in AP, RR, and RO

The minislots serve as reservation slots for the users. Each slot has a duration equal to the maximum propagation time (τ). The nodes are arranged in some sequence in which they may broadcast. If a node wishes to broadcast a packet, it simply sends a signal on the line. If it does not wish to broadcast, it remains silent. Hence, after a duration of τ sec, all the nodes are aware of this particular node's intentions. In effect, the token (i.e., control of the line) is passed to the next user in the form of silence.*

As shown in figure 36, "space" is provided for one packet, followed by a slot of duration τ in which the packet clears the line.

* The scheme is clearly similar to hub polling, in which control passes from user to user [SCHW77], with the major exception that in this case the control is decentralized.

Following Kleinrock and Scholl, the access scheme(s) may be formally characterized as follows:

> The N users are ordered in each slot by the priority rule which characterizes the protocol. For all priority rules (and thus for all protocols) the N users are synchronized as follows in each slot:
>
> 1) If the highest priority user is ready, he need not sense the channel. He synchronizes his packet's transmission as follows:
>
>   (i) At the beginning of the slot, he begins transmission of the carrier (with no modulation).
>
>   (ii) (N-1) minislots later he transmits his packet. Otherwise (if he is idle) he remains quiet until the end of the slot.
>
> 2) If the i^{th} user in priority $(1 < i \leq N)$ is ready, he senses the channel for (i-1) minislots.
>
>   (i) If no carrier is detected after (i-1) minislots then at the beginning of the i^{th} minislot, he transmits his carrier and (N-i) minislots later, he transmits the packet.
>
>   (ii) Otherwise (idle user or carrier detected earlier) he waits for the next slot and the process is repeated (with a possibly different priority order).
>
> Under all protocols, a slot is unused if and only if all users are idle. [KLEI77a, p. 22.1-106]

We also present Kleinrock and Scholl's descriptions of the priority structures which determine the order of transmission.

> In the AP (Alternating Priorities) protocol, the N users are ordered according to a fixed sequence, i=1, ... , N. Then...
>
> 1) Assign the slot to that user (say user i) who transmitted the last packet. If user i is ready,

he transmits a packet in this slot. Otherwise (if there are no more packets from this user),

2) Assign the slot to the next user in sequence (i.e., user (i mod N) + 1).

   (i) If this next user is ready, he transmits a packet in this slot, and in the following slot, operate as above.

   (ii) If this next user is idle, then repeat step 2 until either a ready user is found or the N users have been scanned. In this latter case (all users idle), the slot is unused and in the following slot, operate as above. (The following slot is assigned to user i.) [KLEI77a, p. 22.1-107]

In the RR (Round Robin) protocol, users are assigned slots in cyclic fashion. As in TDMA, each user is pre-assigned one slot in a Round Robin (i.e., cyclic) fashion according to a given sequence of say, (1, 2, ... , N, 1, 2, ...).

1) If the user (say i) to whom the current slot is assigned is ready, he transmits a packet in this slot.

2) Otherwise (user i idle) assign the slot to the next user in sequence (i.e., user (i mod N) + 1).

   (i) If this next user is ready, he transmits a packet in this slot.

   (ii) If this next user is idle, then repeat step 2 until either a ready user is found or the N users have been scanned. In this latter case (all users idle), the slot is unused.

3) No matter who uses the current slot (assigned to user i), the next slot is assigned to user (i mod N) + 1. [KLEI77a, p. 22.1-107].

The RO (random order) protocol assigns priorities to the nodes randomly. Each user generates the same pseudo-random permutation of the digits 1, ... , N in order to determine which node has access to the next slot. A new user is chosen in this manner irrespective of who had use of the last slot.

Kleinrock and Scholl developed models to evaluate the channel capacity (maximum channel utilization) as well as the time delay/throughput trade-off. In addition, they compared the performance of these three protocols and the performance of other access methods (CSMA, TDMA, and polling). A brief summary of these models is followed by an evaluation of the three protocols.

Channel Capacity

Since $N\tau$ sec are lost each frame, the channel capacity C of all three schemes is given by

$$C = \frac{1}{1 + Na} .$$

From this equation, it follows that the capacity of these protocols will be large if either the number of users (N) is relatively small (N = 10) or if the value of a is small (0.001). If for example, a = 0.001. then the capacity will be more than 90 percent for N < 110. More details may be found in KLEI77a.

Packet Delay

All three schemes may be modeled as M/D/1* priority queueing systems with a rest period (corresponding to the reservation slot). The total packet input rate is denoted by λ packets/sec. It is shown in SCHO76 that for equal input rates $\lambda_i = \lambda/N$, i=1, ... , N, the delay is given by

*M/D/1 indicates Exponential arrivals/Deterministic Service/one server.

$$D_i = \frac{1}{2(1 - p)} + 1 \quad \text{for } i=1, \ldots , N,$$

where p = (1 + Na)P is the total normalized input rate (packets/slot). Hence the delay is independent of the protocol chosen in the equal input case.

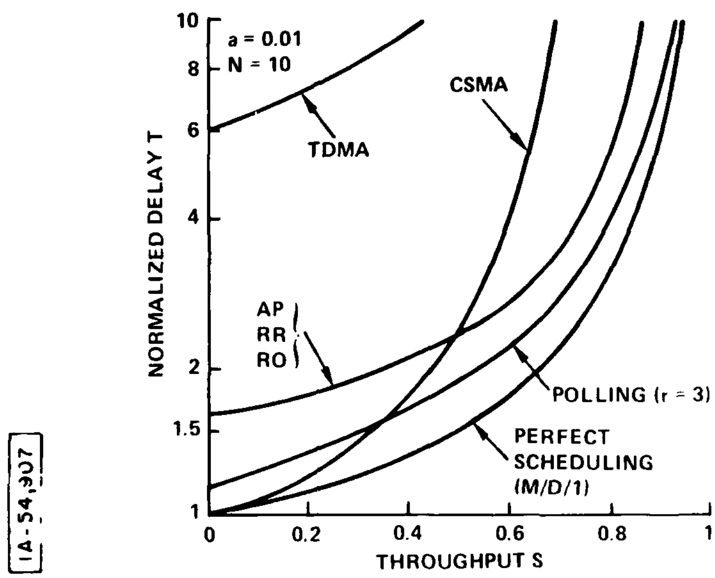
Because the derivation of a delay formula in the case of unequal inputs is difficult, simulation was employed to obtain a better picture of the delay/throughput performance of the three protocols. The conclusion reached as a result of the simulation was that there was very little difference in the mean delay/throughput performance of the three protocols, and that there was only a small difference with respect to the variance.

The delay/throughput performance of the AP, RR, and RO protocols was compared to the performance of several other access schemes -- CSMA, TDMA, and polling. The results are illustrated in figures 37 and 38. For a discussion of the models employed for the various access schemes in obtaining this graph, the reader is referred to SCHO76 or KLEI77a.

Figure 37 illustrates that when there are a small number of users (N = 10) reasonably close together (a = 0.01), AP, RR, and RO perform better than CSMA in a high-traffic environment. For a larger number of users (50), the performance profile changes a great deal, as indicated by figure 38.

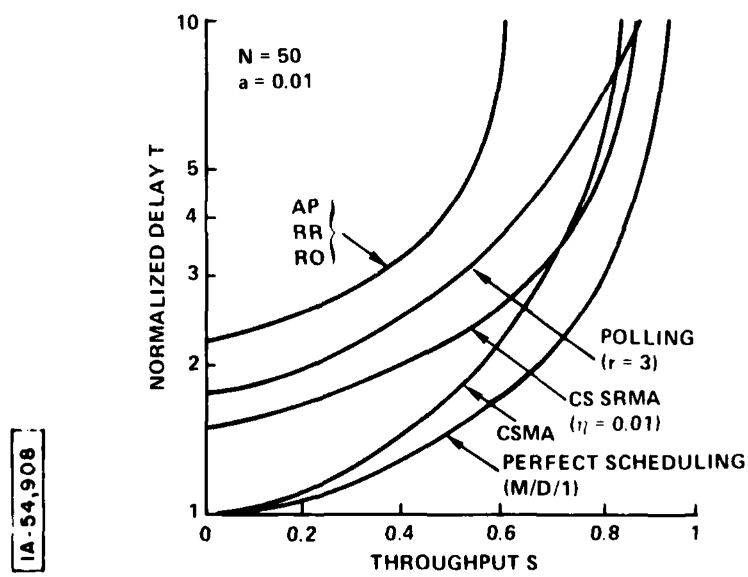
In summarizing the results of their analyses, Kleinrock and Scholl make the following points:

1. For a small number of users, AP, RR, and RO provide a good channel capacity and a delay/throughput performance comparable to polling or CS SRMA (figure 37).



Based on KLEI77, p. 22.1-108.

Figure 37. Packet Delay Versus Throughput (N = 10, a = 0.01)



Based on KLEI77, p. 22.1-108.

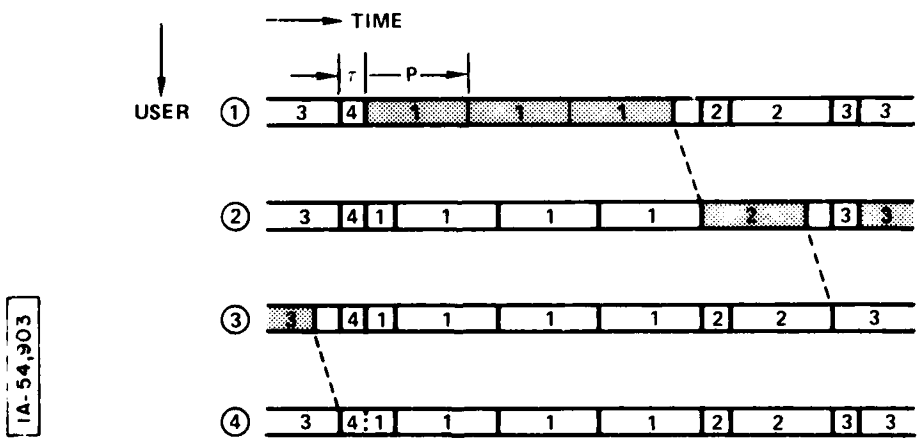
Figure 38. Packet Delay Versus Throughput (N = 50, a = 0.01)

2. For sufficiently small a (e.g., a = 0.001), that is, when the users are sufficiently close together, the number of users can be reasonably large (50) without significant performance degradation. For larger a (e.g., a = 0.01), the performance degrades with the number of users, as indicated in figure 37.

It is clear from examination of the three protocols that a certain amount of channel time is consumed during the reservation portion of the frame. In order to reduce this time, the authors developed the MSAP protocol. The feature which distinguishes this protocol from AP, RO, and RR is that a user begins his transmission in a packet slot immediately after indicating that he has something to transmit in a reservation slot. The users are assigned a priority structure as with the preceding three protocols. The protocol obeys the AP structure (i.e., the last user to transmit retains control of the channel if his buffer is not empty. More formally, the protocol operates as follows. Assume that user i is the last user to transmit a packet.

> By carrier sensing, at most one (mini-)slot later, all users detect the end of transmission of user i (absence of carrier); in particular, so does the next user in sequence (user (i mod N) + 1. Then
>
> (i) either: User (i mod N) + 1 starts transmission of a packet; in this case, one slot after the beginning of his transmission, all others detect the carrier. They wait until the end of this packet's transmission and then operate as above.
>
> (ii) or: User (i mod N) + 1 is idle; in this case, one slot later, all other users do not detect the carrier; they know that it is the turn of the next user in sequence, i.e., user (i mod N) + 2 and operate as above. [KLEI77a, p. 22.1-109].

Figure 39 illustrates MSAP's operation. In this example, two slots after user 3's transmission, user 1 gains control of the line as he detects that user 4 is idle. User 1 then transmits three packets, and is followed in turn by users 2 and 3. This example underlines the fact that the overhead for MSAP is significantly smaller than that of the AP, RR, or RO protocols. The only time lost on the channel under MSAP is one minislot due to switch-over from one user to another. This is in contrast to the family of three protocols, in which N (number of users) minislots are lost at each packet transmission.



Based on KLEI77, p. 22.1-109.

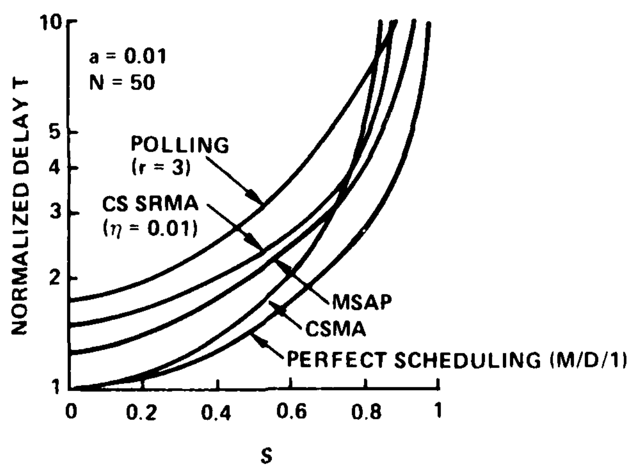
Figure 39. MSAP, Four Users

In order to obtain an equation for the time delay, we note the similarity of MSAP to roll-call polling [SCHW77]. The difference between the two schemes is due to the larger change-over time between two users in polling. In roll-call polling, this change-over time equals the polling message transmission time (≥ 1 slot), plus twice the propagation time between the user and the central station. Kamheim and Meister's results on roll-call polling may then be applied by simply setting the polling time equal to 1 slot in their equation [KONH74]. The expected packet delay T is then given by:

$$T = 1 + \frac{S}{2(1 - S)} + \frac{a}{2}(1 - \frac{S}{N})(1 + \frac{N}{1 - S}),$$

where S is the throughput, measured in packets generated per transmission time P.

By way of making a comparison between MSAP and other access methods, Kleinrock and Scholl provide the following graph (figure 40). In this figure, it is assumed that there are 50 users (N = 50) and that a = 0.01.



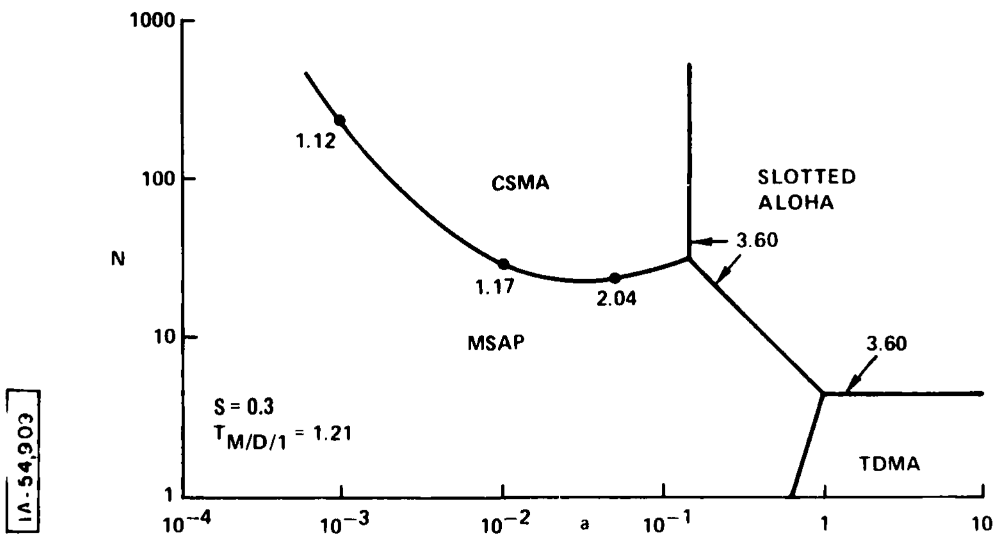
IA-54,900

Based on KLEI77, p. 22.1-110.

Figure 40. Time Delay Versus Throughput

As figure 40 illustrates, CSMA performs better than MSAP at light traffic loads, while MSAP performs better at higher loads. Because the difference in performance of the two protocols at light loads will not be great, MSAP is a better overall choice than CSMA.

It is clear from the comparisons made thus far that the appropriate choice of an access protocol is very much dependent on the environment in which it is used. For example, under light traffic conditions with a large number of users, the preceding conclusion is reversed. Hence, the authors provide a series of graphs for comparing the various protocols. The graphs display the best protocol (lowest time delay) at a given throughput for different values of N and a (a is the normalized propagation delay). The first graph (figure 41) displays this trade-off for the light traffic case (S = 0.3).

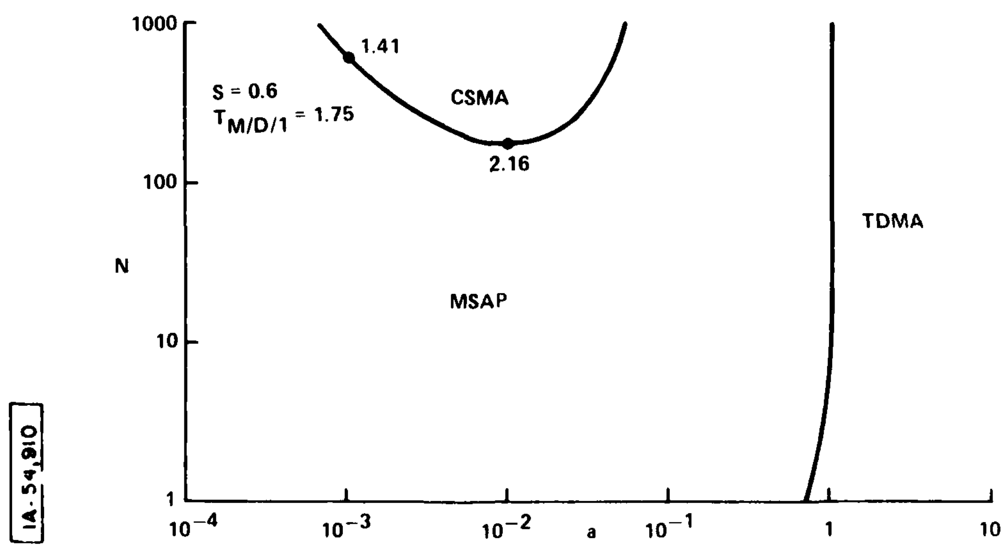


Based on KLEI77, p. 22.1-110.

Figure 41. N Versus a (S = 0.3)

We note that four regions are delimited in which each of the protocols indicated provides the lowest delay. For a large population of users, random access techniques provide the best performance. As the value of a increases, the performance of CSMA degrades below that of slotted ALOHA. MSAP is the preferred choice for a small- to medium-size population that is not widely dispersed (a < 0.6), while TDMA provides the best performance for a small population that is widely dispersed.

Figure 42 illustrates the heavy traffic case of S = 0.6. Note that at heavier traffic loads, MSAP is to be preferred over CSMA even for a large number of users. When the users become widely dispersed (a ≈ 1, i.e., propagation time approximately equal to transmission time) TDMA is the preferred choice. The numbers on the boundary lines of the figures are the ratio of the time delay of the protocol to that of perfect scheduling, i.e., an M/D/1 queue.



Based on KLEI77, p. 22.1-110.

Figure 42. N Versus a (S = 0.6)

MLMA

MLMA, short for Multi-Level Multi-Access Protocol, was developed at IBM's Zurich Laboratories by E. Rothauser and D. Wild. They describe this protocol in a paper, "MLMA - A Collision-Free Multi-Access Method," [ROTH77], which forms the basis for this discussion.

The scheme is similar to Kleinrock and Scholl's AP, RR, and RO protocols in that it breaks a time frame into reservation minislots and message slots. The presumed environment for MLMA is one in which the number of users is large, but the average number of requests for channel time is small. Following the reservation frame is a message frame that consists of a variable number of packet slots, one for each reservation made in the reservation frame.

Figure 43 is a diagram of the message frame. In its simplest implementation, users request packet slots via a "one out of N" code, where N is the number of users -- i.e., each user sets his bit to one in order to request packet slots. As this scheme clearly becomes inefficient for large N, the authors extend their scheme to a multi-level code, which operates as follows. Let S be the base of a number system (e.g., S = 10), then any requestor's address R_j may be represented as:

$$R_j = \sum_{j=0}^{m-1} C_j S^j.$$

For example, $101 = 1\ (10)^2 + 0\ (10)^1 + 1\ (10)^0$.

The coefficients C_j (= 1, 0, 1 in the above example) are transmitted in each request subslot. Rothauser and Wild give an example in which they claim that for N = 1000 terminals, and S = 10, the length of the request slot can be shrunk from 1000 bits to 30 bits. In order to

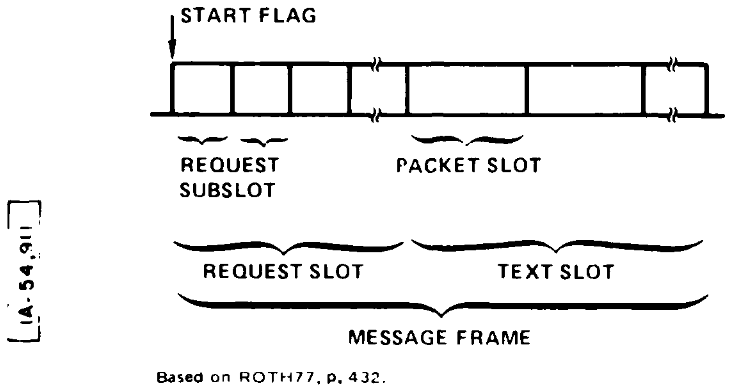


Figure 43. Structure of the Message Frame

circumvent the problem of multiple requests (e.g., from terminals 013. 016, 522, and 579), the number of request subslots actually employed must be increased. Figure 44 (taken from ROTH77) illustrates both the problem of multiple requests and the authors' solution to it.

We assume requests from terminals 013, 016, 522, and 579 as above. Figure 44 illustrates the process of identifying the requesting terminals (a multilevel tree).

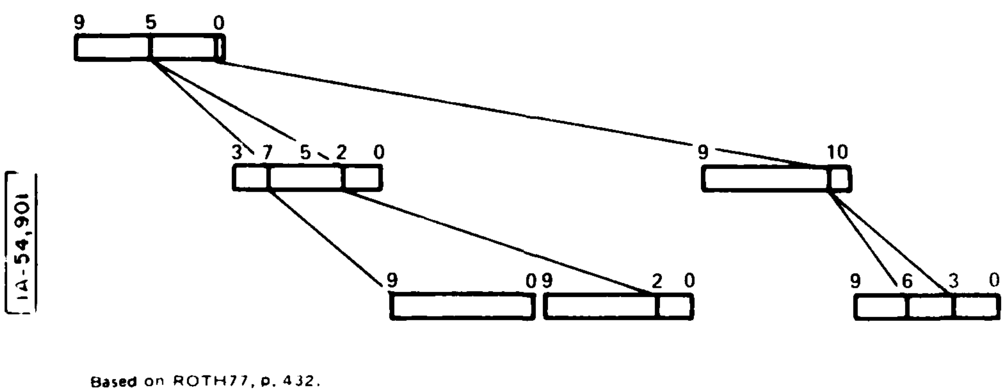


Figure 44. Three-Level Tree of Requestor Addresses

After the first request subslot, all users are aware that several terminals which have an address starting with either 0 or 5 are requesting space to broadcast (the exact number of requesting terminals is unknown at this point). The terminal(s) that have inserted the highest bit (5) in the first subslot are then requested to place their second bit in the next request subslot. Bits 7 and 2 are then set as a result of this request. Repeating this procedure, the next two subslots are employed to differentiate between the last two digits of the addresses 572 and 579. As indicated on the diagram, only three slots are required to differentiate between terminals 013 and 016.

Assuming no multiple requests, one may obtain a simple relationship between the number of terminals N, the number of levels in the system N, and the length of the request subslot X, as follows:

$$N = X^n .$$

Under the same hypotheses the channel utilization CU may be computed as follows:

$$CU = \frac{P}{n\lceil \sqrt[n]{N} + P} ,$$

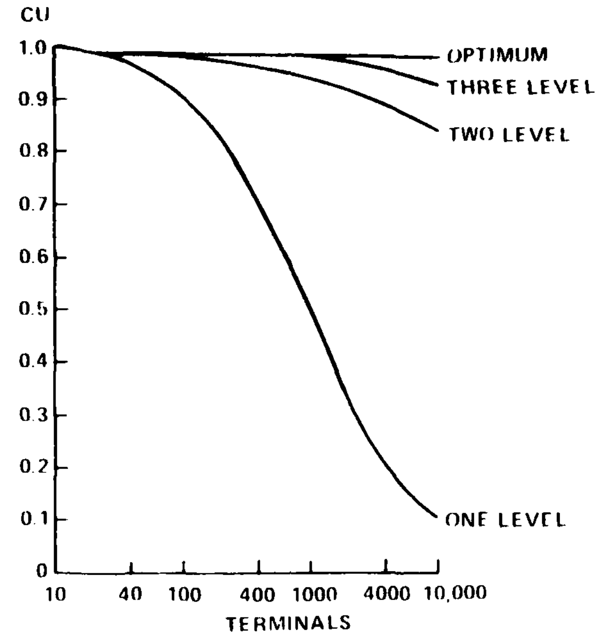
where

P = packet length,

$\sqrt[n]{N}$ = X, the number of bits in a request subslot,

$\lceil$ = next largest integer.

The expression in the denominator includes $n \cdot \lceil \sqrt[n]{N}$ as the overhead for each message slot. On the basis of this formula, Rothauser and Wild obtained curves such as those shown in figure 45 in order to determine the appropriate number of levels.



Based on ROTH77, p. 433.

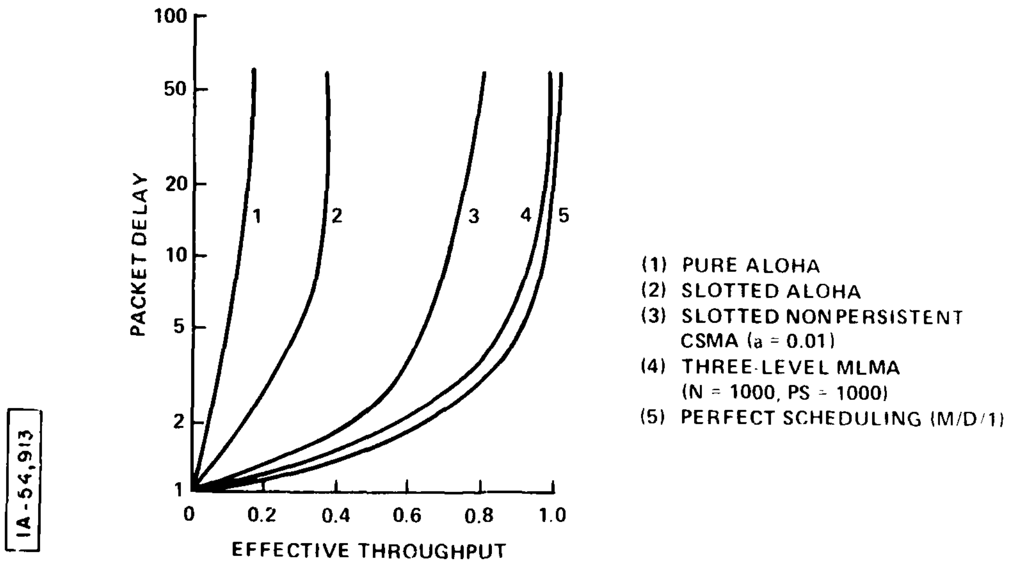
Figure 45. Channel Utilization Versus Number of Terminals (Packet Size = 1000 Bits)

In deriving a time delay equation, the authors simplify the problem by neglecting (1) the size of the request slot on the grounds that it is smaller than the message slot, and (2) prioritization among terminals. The result of this simplification is ideal -- an M/D/1 queueing system, in which the average packet delay is given by

$$W = \frac{\rho}{2(1 - \rho)} ,$$

with ρ as the offered traffic.

The authors also compare their protocol with several others via the time delay/throughput trade-off displayed below (figure 46).



Based on ROTH77, p. 435.

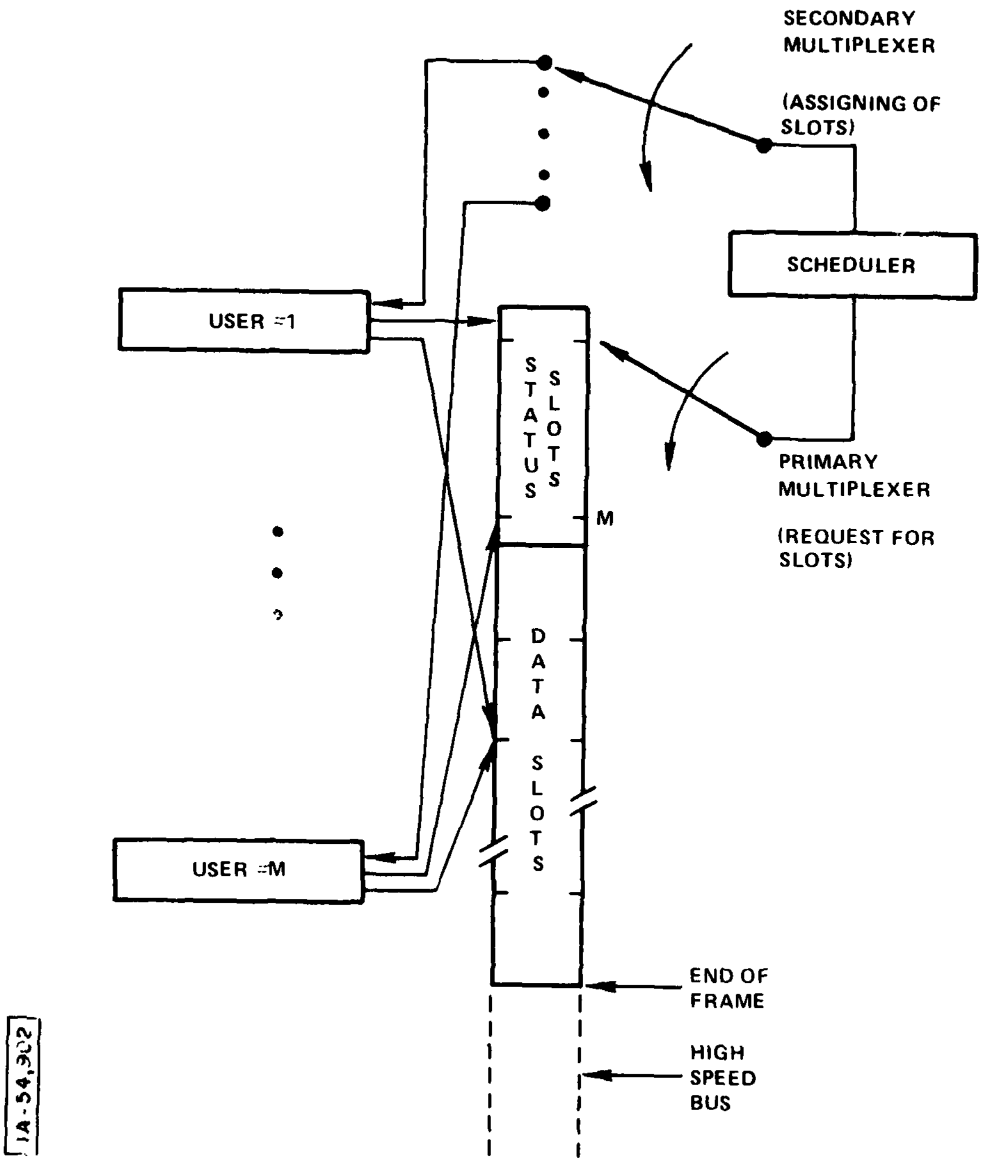
Figure 46. Packet Delay Versus Throughput

GSMA

GSMA (Global Scheduling Multiple Access) was developed by Jon Mark at IBM's Watson Laboratories [MARK78].

The feature that distinguishes GSMA from the other protocols introduced so far is the fact that it is centrally controlled. As with the systems already discussed, time frames are broken up into reservation and message slots, (called status and data slots by Mark). A diagram of a typical message frame is presented in figure 47.

The i^{th} user is assigned a status slot s_i bits long, which he uses to indicate how many packets he has in his buffers, as well as his priority status. Mark assumes that the status slot is used solely to provide information as to the number of packets awaiting transmission.



Based on MARK78, p. 1343.

Figure 47. GSMA Slot Assignment

Hence, each user can report a total of $2^{s_i} - 1$ packets to the central scheduler. The scheduler accepts reservation requests, and informs each user of its place in the broadcast line.

The system, as Mark points out, may be thought of as consisting of a single server serving each user in cyclic fashion. At the start of the j^{th} frame, the user reports the number of packets that have arrived during the $(j - 1)^{st}$ frame in his status slots, in addition to sending the packets he requested space for in the $(j - 1)^{st}$ frame.

GSMA operates with two different formats (for the status slots) -- GSMA(s) and GSMA(m). In GSMA(s), each user has a status slot of one bit, while in GSMA(m), each user has an m-bit status slot.

Mark has developed expressions for the channel utilization as well as for the time delay of GSMA(m). He assumes independent arrival processes at each node (and for each individual frame) that are governed by identical distributions of the form

$$f_i(k) = P(x_{i,j} = k),$$

where $x_{i,j}$ is the number of data units arriving at the i^{th} user during the j^{th} frame. Then an expression for $n_{i,j}$, the number of packets that remain to be transmitted by the i^{th} user during the j^{th} frame is given by

$$n_{i,j} = n_{i,j-1} - q_{i,j-1} + x_{i,j} \quad i=1, \ldots, M,$$

where $q_{i,j-1}$ is the number of packets transmitted during the $(j - 1)^{st}$ frame. A constraint upon this equation is that

$$q_{i,j} \le n_{i,j} \quad i=1, \ldots, M, \; j=1, \ldots.$$

This equation of the processes' evolution is unfortunately non-Markovian in nature (the transition probability is dependent upon the history of the process). Hence, using Kendall's approach [KEND53], Mark employs an embedded Markov chain to develop his queueing models.

Letting τ_j, $j=1, \ldots, \infty$ represent the instants in time at which the frames commence, Mark defines the cycle time t_j to be

$$t_j = \tau_j - \tau_{j-1} \quad j=1, \ldots, \infty.$$

Since $n_{i,j+1}$ depends upon $n_{i,j}$, it follows that $t_{i,j+1}$ depends upon $t_{i,j}$. Noting that Konheim investigated a similar system [KONH76], and discovered the influence of this dependency to be negligible, Mark makes the assumption that the set of cycle times t_j, $j=1, \ldots, \infty$ is independent, and therefore the set (τ_j) forms an embedded Markov process.

In developing an expression for the channel utilization, the following expression for the average number of customers $\overline{L}$ is used:

$$\overline{L} = \sum_{i=1}^{M} \overline{q}_i,$$

where q_i is the mean number of packets reported by the i^{th} user, given by

$$\overline{q}_i = \sum_{k=0}^{2s_i-1} k \; Pq_i^{(k)}.$$

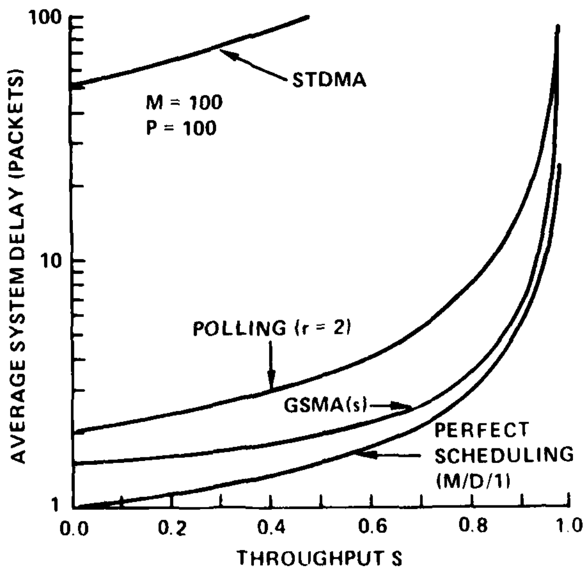
$Pq_i^{(k)}$ is the probability that $q_i = k$.

The channel utilization for GSMA(m) is then given by

$$CU_{GSMA(m)} = \frac{\bar{L} \cdot P}{\sum_{i=1}^{M} s_i + \bar{L} \cdot P} .$$

Expressions for the i^{th} user's mean queue size and mean time delay are also derived. (As usual, the expression for the mean time delay is a consequence of Little's theorem.) As these expressions are rather cumbersome, the reader is referred to Mark's paper.

Comparing GSMA(s) time delay/throughput performance to that of Synchronous Time-Division Multiplexing (STDMA), polling, and finally to perfect scheduling, yields figure 48. In developing these curves, Mark assumed equal arrival rates at all of the nodes. The curves for STDMA and polling are based upon analytical results developed by Lam [LAM76] and Konheim [KONH74].



IA-54,914

Based on MARK78, p. 1349.

Figure 48. Packet Delay Versus Throughput

DYN

Kleinrock suggests the Dynamic Reservation scheme (DYN) in his paper, "Performance of Distributed Multiaccess Computer-Communications Systems" [KLEI77b]. His purpose in developing this access scheme was two-fold:

1. To create a scheme capable of good performance at high traffic levels (better, for example, than MSAP).

2. To create a scheme whose behavior does not depend heavily on how widely the users of the system are dispersed.

His description and analysis of this scheme is cursory, and is, in his own words, "basically intended to illustrate that dynamic control schemes do exist which have rather efficient behavior in the heavy traffic" [KLEI77b, p. 550]. Hence, only a brief summary of the scheme and the attendant analysis will be presented.

Under this protocol a channel of capacity C is divided into a reservation channel of capacity C_R and a data channel of capacity C_D. The capacity C of the channel is therefore given by $C = C_D + C_R$. Each user sends a brief message over the reservation channel and thereby gains control of the data channel until he is finished. Service on the data channel is first-come first-served. The reservation channel operates in a TDMA mode. Since it is a broadcast channel, any activity on the channel indicates a desire to transmit, and a one-bit message is all that is required to reserve the data channel.

Kleinrock develops an expression for a packet's average time delay, defined as the average time from the time the packet is generated until it is successfully received. The time delay is measured in packet transmission times -- i.e., one divides the actual time delay by $\frac{b}{C}$, where b is the packet length, and C is the channel capacity. The time delay is also defined as a function of load S on the channel, equal to $\frac{\lambda b}{C}$ where λ is the sum of the arrival rates at the users.

The formula developed is the sum of three terms:

1. The reservation time required, shown to be equal to $\frac{(M + 2)C}{2bC_R}$, where M is the number of users, b is the packet length in bits, and C_P is the capacity of the reservation channel.

2. The transmission delay on the data channel shown to result from an M/D/1 queueing system [SCHO76]. This is equal to $\frac{(2 - S_D)C}{2C_D(1 - S_D)}$, where S_D is the load on the data channel and C_D is the capacity of the data channel.

3. The propagation delay a on the channel.

The normalized response time, $T_{DYN}^{(s)}$ is therefore given by

$$T_{DYN}(S) = \frac{(M + 2)C}{2bC_R} + \frac{(2 - S_D)C}{2C_D(1 - S_D)} + a.$$

Expressions are developed for C_R and C_D by first developing an upper bound for the load placed on the data channel (denoted by 0) and then employing the definitions $C = C_R + C_D$, $S_D = \frac{SC}{C_D}$ and $S = \frac{\lambda b}{C}$. The expression developed for σ is

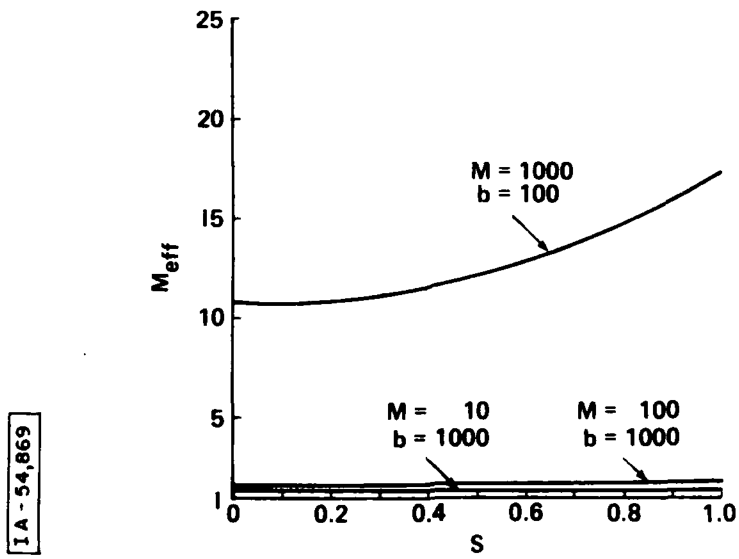
$$\sigma = \frac{2\lambda M(1 - S_D)}{C_R S_D} .$$

The expressions for C_R and C_D are substituted into the above expression for $T_{DYN}(S)$, resulting in

$$T_{DYN}(S) = \frac{1}{2(1 - S)} \left(\frac{\sigma b + 2M}{\sigma b}\right) \left[\left(\sigma \frac{M + 2}{2M}\right) + \frac{(2 - S)\sigma b + 2MS}{\sigma b + 2MS}\right] + a.$$

This value optimizes $T_{DYN}(S)$ because $S \rightarrow 1$ (the heavy traffic case) is obtained by first taking the limit of the above expression as $S \rightarrow 1$, and then differentiating the result with respect to σ. To obtain the heavy traffic behavior of the protocol, it suffices to substitute the value obtained for $\sigma = \frac{2M}{\sqrt{b(M + 2)}}$ into $T_{DYN}(S)$ and take the limit of $T_{DYN}(S)$ as $S \rightarrow 1$. This yields $\lim_{S \rightarrow 1} 2(1 - S)T_{DYN}(S) = \left(1 + \sqrt{\frac{(M + 2)}{b}}\right)^2$.

From this expression we note that the heavy traffic behavior of DYN is very good, as illustrated by figure 49. M_{eff} in this figure



Based on KLEI77b, p. 551.

Figure 49. Performance of the Dynamic Reservation Scheme

is the ratio of $T_{DYN}(S)$ and $T_{M/D/1}(S)$ (perfect scheduling) and M is the number of users.

The performance of this dynamic scheme (as illustrated by figure 46) is excellent unless the number of users (M) far exceeds the packet length. It also outperforms MSAP provided that a critical value for a is not exceeded [SCH076]. DYN and MSAP, as well as other protocols, are compared in the summary section of this paper.

THE URN SCHEME (AN ADAPTIVE SCHEME FOR MULTIPLE ACCESS)

The Urn scheme is an adaptive scheme for distributed multiple access to broadcast systems. It varies from slotted ALOHA at light loads, to an asymmetric scheme at intermediate loads, and finally results in time division multiple access (TDMA) at heavy loads. It was developed at UCLA by Kleinrock and Yemeni and reported in "An Optimal Adaptive Scheme for Multiple Access Broadcast Communications," [KLEI78].

Operating under a distributed control environment, the scheme assumes that each user knows how many users are busy at the beginning of a time slot. Methods for gathering this information will be summarized later. We commence with a summary of the access scheme <u>assuming that this information is available to each user</u>.

An example of how the scheme works serves to illustrate the general solution to the problem. We assume two users (N = 2), and first consider the case in which one is busy (n = 1). If both users are given access rights (symmetric transmission policy) at the beginning of a slot, then one will broadcast and the other will not.

A symmetric policy is not always ideal, as is shown by the case of two users, both busy. Giving each equal transmission probabilities (0.5) results in a 0.5 throughput, as only two out of the four possible outcomes of this policy can produce a successful transmission. Hence, it is better to allow only one of the users to transmit with probability 1.

Yemeni, in his thesis, "On Channel Sharing in Discrete-Time Packet Switched, Multiaccess Broadcast Communication" [YEMI78] proves that the optimal strategy for assignment of access rights is asymmetric -- i.e., several users should be allowed full access rights, while the remainder should be allowed none. This is in contrast to symmetric policies, in which each user has equal access rights.

The model upon which Yemeni's scheme is built is the classic probability situation of drawing colored balls from an urn (hence the name of the scheme). The traditional colors of black and white are chosen to represent the busy and idle users. The urn contains N (number of users) balls of which n are assumed to be busy. If k balls are drawn from the urn, then the probability of a successful transmission is the probability of drawing one black ball from the urn. This is given by the hypergeometric distribution as

$$\frac{\binom{k}{1}\binom{N-k}{n-1}}{\binom{N}{n}}\ .$$

It can be shown that this probability is maximized when $k = \left\lfloor \frac{N}{n} \right\rfloor$, where $\lfloor x \rfloor$ is the integer part of x. This model is generalized in YEMI78.

When there is only one busy user ($n = 1$), the above formula implies that $k = N$ -- that is, all the users have full access rights (although only one user will make use of them). This is essentially slotted ALOHA. As the load (n) increases, k decreases, allowing fewer users access to the channel. When $n > \frac{N}{2}$, only one user is allowed access ($k = 1$), and we have a TDMA scheme (random TDMA if the sampling is random, and round-robin TDMA if it is without repetition). Hence, when traffic is light, a small amount of channel capacity is lost due to collisions as the scheme operates in a slotted ALOHA mode. However, as the traffic load increases, collisions are gradually eliminated, thus maintaining a high channel utilization.

Kleinrock and Yemeni suggest a scheme for estimating the number of busy users. The scheme consists of a binary erasure reservation subchannel that may be implemented via a minislot at the beginning of each data slot. An idle user that turns busy sends a message of a few bits in this slot. Each user is assumed able to detect the presence or absence of a reservation as well as an erasure (resulting from colliding messages). Simulation studies show that assuming there are always two users involved in a collision is an effective way of estimating the number of busy users.

The specific k users that actually get access rights at any given instant can be selected by a variety of means. For example, the same random number generator that selects k out of N users may be employed by each user. The authors suggest a scheme they call a "window scheme," which reduces k when collisions occur.

Both analytic and simulation studies were conducted on the performance of the urn scheme. Its performance was compared to that of:

1. Optimally controlled slotted ALOHA (transmission probability = $\frac{1}{n}$).
2. Random TDMA.
3. Perfect scheduling.

Assumptions for both models include:

1. Arrival distributions are time-independent and arrivals occur independently of the slots.

2. Service mechanisms are time-independent, and are independent of the arrival process.

In order to determine an expression for the time delay of the system, a model was first constructed for the number of busy users in the system. Its transition equation is represented below.

$$(\pi_0, \pi_1, \cdots \pi_N) = (\pi_0, \pi_1, \cdots \pi_N) \begin{bmatrix} \alpha_1^0 & \alpha_2^0 & . & . & . & . & . & \alpha_{N+1}^0 \\ \alpha_0^1 & \alpha_1^1 & . & . & . & . & . & \alpha_N^1 \\ 0 & \alpha_0^2 & . & . & . & . & . & . \\ & 0 & & & & & & . \\ & & . & . & . & . & . & . \\ & & 0 & & & \alpha_0^N & & \alpha_1^N \end{bmatrix},$$

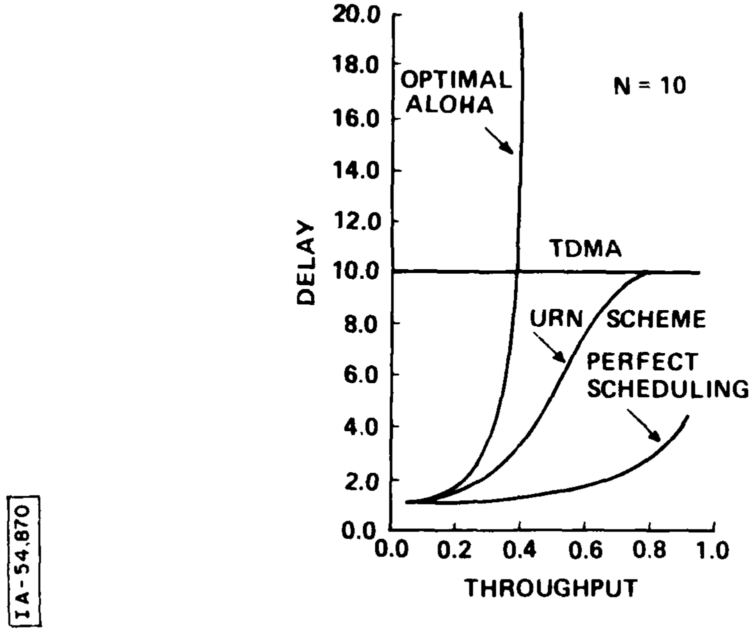
where

π_n = equilibrium probabilities.

α_i^n = transition probabilities.

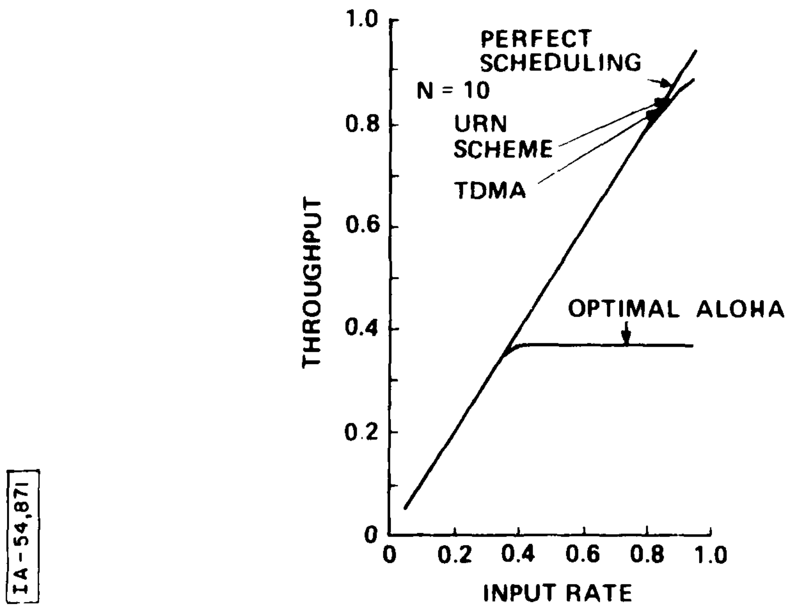
Solving these equations and then employing Little's theorem, the authors obtain the following curves for the delay/throughput performance (figure 50) and for the input/throughput rate performance (figure 51).

Simulation studies have provided confirmation for these graphs.



Based on KLEI78, p. 7.2.4.

Figure 50. Delay-Throughput Comparative Analysis



Based on KLEI78, p. 7.2.4.

Figure 51. Input-Throughput Analysis

SECTION 4

MODELING OF BUS NETWORKS - CONCLUDING REMARKS

In this section, to the extent possible, the various bus network access schemes discussed in this paper are compared. The performance measure employed is the time delay versus throughput trade-off. In addition, the models used in evaluating the performance of the access protocols are discussed.

COMPARISON OF ACCESS PROTOCOLS

Any comparison of access schemes must attach a great deal of importance to the environment in which the protocol is to be employed. Kleinrock, in an important paper, "Performance of Distributed Multi-Access Computer-Communications Systems" [KLEI77b], mentions basic parameters that serve to characterize a distributed local broadcast communications system. They are:

- The number of users, M, attached to the system.
- The loads, S_m, generated by each user.
- The (normalized) propagation delay, a, which serves as a measure of the dispersion of the users of the system.
- The total system load, $S = \sum_m S_m$, placed upon the entire system.

Kleinrock's system description may be simplified by noting that the last parameter is obtained from knowledge of the first two sets of parameters.

Kleinrock goes on to note that nature exacts a price for distributed control in the form of collisions, idle capacity or control overhead. Table 1 [KLEI77b] illustrates this price for three categories of systems.

Table 1

The Price for Distributed Sources

	Collisions	Idle Capacity	Overhead
No Control (e.g. ALOHA)	Yes	No	No
Static Control (e.g. FDMA)	No	Yes	No
Dynamic Control (e.g. Reservation Systems)	No	No	Yes

Based on KLEI77b, p. 548.

A systems designer's problem is therefore to decide which price (or combination of prices) is appropriate to pay for his particular environment (as characterized by Kleinrock's three parameters). The performance measure apparently most appropriate for making this decision is the time delay versus throughput trade-off* made for differing values of M, a, and S. Charts similar to the ones presented in the

* Stability is an important consideration for random access protocols. However, dynamic control procedures have been developed which produce close-to-optimal, stable channel delay-throughput performance [LAM74 or KLEI75].

discussion of MSAP (figures 41 and 42) should form the basis of a decision as to which protocol should be employed for a given system. Figures 41 and 42 illustrate how dependent the choice of the appropriate protocol is upon the particular environment (characterized by the parameters S, M, and a).

Kleinrock compares a number of multi-access protocols in his paper. In doing so, he first defines a "new" system parameter, M_{eff}, by the following formula:

$$M_{eff} = \frac{T_x(S)}{T_{M/D/1}},$$

where

$T_x(S)$ is the time delay of system x at load S.

$T_{M/D/1}$ is the time delay for the ideal deterministic system.

One may view this ratio as equal to an effective number of system users via the following observations [KLEI74], referred to as the "scaling law" with $\frac{C}{M}$. The scaling law states that if we compare the time delay for the following two categories of systems:

- M systems, each with capacity $\frac{C}{M}$ and input rate $\frac{S}{M}$.

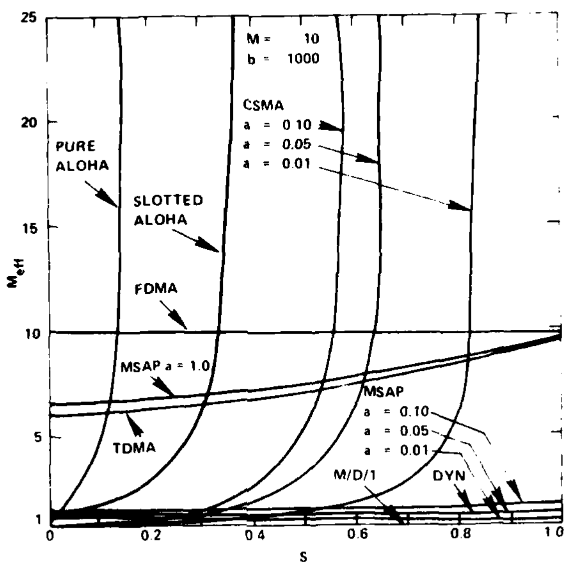
- A single system handling total input rate S with total capacity C .

Then the second system would have an average time delay equal to $\frac{1}{M}$ of the time delay of the first system.

A corollary of this law is that an FDMA system has a time delay that is M times as large as the deterministic M/D/1 system. As an example of this law, we take a system of 100 users each provided with an average 150 μsec time delay. If an M/D/1 system can provide each user with a 3 μsec time delay, then in effect the M/D/1 system provides 50 users (= M_{eff}) with the same 150 μsec time delay as the (proposed) system provides one user.

This new parameter* M_{eff} provides a reduced system description -- M_{eff} users provided load S, and distributed at distance a.

Employing M_{eff} as a system parameter, Kleinrock presents the time delay versus throughput trade-off for a wide range of systems. Three separate cases are presented in figures 52, 53, and 54:

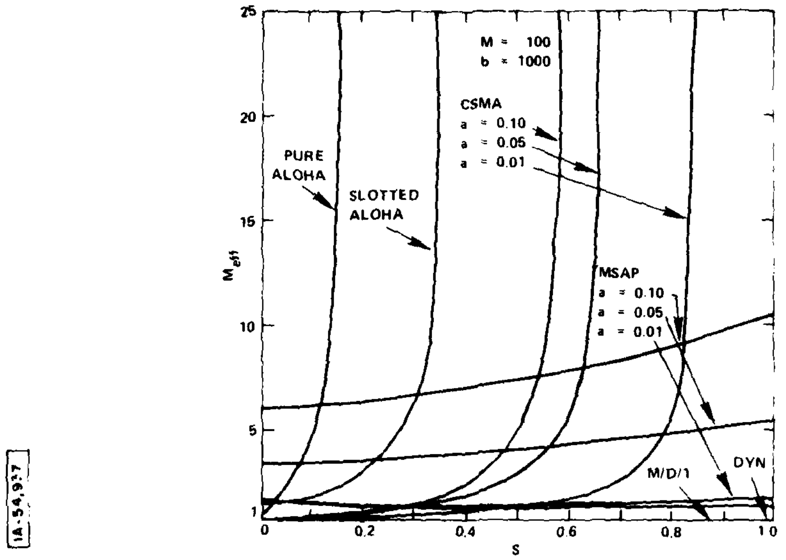


1A-54,936

Based on KLEI77b, p. 551.

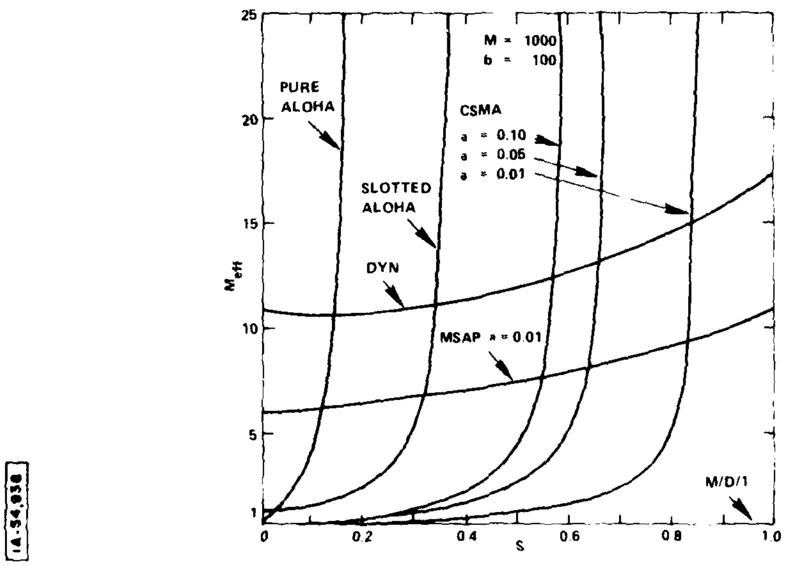
Figure 52. Response Time Ratios (M=10, b=1000)

* Since loads are not uniformly distributed in a system, one must allocate the system capacity in such a way as to minimize the time delay.



Based on KLEI77b, p. 551.

Figure 53. Response Time Ratios (M=100, b=1000)



Based on KLEI77b, p. 552.

Figure 54. Response Time Ratios (M=1000, b=100)

- 10 users with a packet length b of 1000 bits (figure 52)
- 100 users with a packet length b of 1000 bits (figure 53)
- 1000 users with a packet length b of 100 bits (figure 54)

DYN stands for the Dynamic Reservation scheme discusses previously. These figures point out (among other things) that of the schemes compared, either MSAP or DYN is best in the heavy traffic end.

Figures 52, 53, and 54 do not afford a definitive comparison of broadcast access schemes. A number of protocols are not included. Among these are:

- LWT protocol
- prioritized CSMA (e.g. Hyperchannel)
- Urn protocol
- MLMA
- GSAP

It might also be desirable to portray a more detailed picture of the effect of the parameter a on these access protocols (as is done for the graphs illustrating MSAP).

PERFORMANCE MODELS OF ACCESS PROTOCOLS

Queueing and simulation models have been created for all the access protocols examined in Section 3. In general, there has been good agreement between both categories of models. While the queueing formulas produce results more quickly than do the more detailed simulations, the advantages of simulation are clearly shown in two cases:

- In the Hyperchannel simulation studies conducted at Lawrence Livermore Laboratories, the authors pointed out a number of problems caused by the interaction of the user-level protocols with the access-level protocols. These effects could not have been protrayed by queueing formulas that only modeled the access-level protocols. Queueing models of the higher level protocols were not developed, in all likelihood, due to the complexity of the interaction of the two protocol levels.

- The time delay formulas developed by Tobagi for his CSMA protocols proved difficult to optimize. Hence, he employed simulation to obtain a time delay/throughput trade-off.

The conclusion to be reached, therefore, is that both types of models are necessary in conceptual phase modeling. Queueing models may be used to rapidly evaluate design alternatives and discard inappropriate ones, while simulation models may be used for more detailed evaluation of selected designs. There is also benefit in being able to to compare the results produced by the two types of models.

In attempting to develop a unified queueing model for bus networks, Kleinrock has developed a good three-parameter approximation for a great number of the time delay formulas [KLEI77b]. His generic formula for time delay as a function of throughput (S) is:

$$T(S) = A \frac{Z - S}{P - S},$$

where

Z is the zero of the function,

P is the pole of the function, and

A is a scalar multiplier.

Table 2 [KLEI77b], indicates the appropriate values for parameters in four access schemes.

Table 2

Parameter Values

Access Method	Z	A	P
FDMA	2	$\frac{M}{2}$	1
TDMA	$\frac{(2 + M)}{2}$	1	1
M/D/1	2	$\frac{1}{2}$	1
MSAP	$\frac{(2 + a(M + 1))}{(1 + a)}$	$\frac{(a + 1)}{2}$	1

Based on KLEI77b, p. 548.

The ZAP approximation may also be employed on the ALOHA and CSMA schemes by fitting the values of the parameters.

Little work has been done on the modeling of a complete network employing a bus architecture (i.e., modeling of the nodes as well as the communications system). A potentially effective approach to the modeling of such a system would be to use Jackson's open queueing networks to model the nodes [KLEI76], and to use the appropriate time delay formula to model the performance of the bus itself. The ZAP approximation might be effectively employed here.

In spite of the quick results to be obtained by such an approach, there is also danger resulting from a lack of detail. Hence, one might construct a two-tiered model. The first level, consisting of the above queueing model, would be suitable for a rapid, time delay versus throughput analysis. The second level would consist of a simulation of the bus communications network, coupled to a (Jacksonian) model of the nodes as an open network of queues.

REFERENCES

SECTION 2. RING NETWORKS

ANDE72 R. R. Anderson et al., "Simulated Performance of a Ring-Switched Data Network," IEEE Trans. on Communications, COM-20, 3 (June 1972), pp. 576-591.

ANDE75 G. A. Anderson and E. D. Jensen, "Computer Interconnection Structures: Taxonomy, Characteristics, and Examples," Computing Surveys, Vol. 7, No. 4, December 1975, pp. 197-213.

AVI63 B. Avi-Itzhak and P. Naor, "Some Queueing Problems with the Service Station Subject to Breakdown," Oper. Res., Vol. 2, No. 3, 1963, pp. 303-320.

BABI77 G. A. Babic et al., "A Performance Study of the Distributed Loop Computer Network (DLCN)," Proc. of Computer Networking Symposium, December 1977, pp. 66-75.

BASK75 F. Basket et al., "Open, Closed, and Mixed Networks of Queues with Different Classes of Customers," JACM, Vol. 22, April 1975, pp. 248-260.

CARS77 R. T. Carsten et al., "A Simplified Analysis of Scan Times in an Asymmetrical Newhall Loop with Exhaustive Service," IEEE Transactions on Communications, COM-25, 9 (September 1977), pp. 951-957.

FARM69 W. D. Farmer and E. E. Newhall, "An Experimental Distributed Switching System to Handle Bursty Computer Traffic," Proc. of the ACM Symposium on Problems in the Optimization of Data

Communications Systems, Pine Mountain, Georgia, (October 1969), pp. 1-34.

FRAN78 A. Franck, communication dated March 27, 1978.

HAYE71 J. F. Hayes and D. N. Sherman, "Traffic Analysis of a Ring Switched Data Transmission System," The Bell System Technical Journal, 51, 6 (July - August 1972), pp. 2947-2977.

JACK63 J. R. Jackson, "Jobshop - Like Queueing Systems," Management Science 10, No. 1, 1963, pp. 131-142.

KAYE72 A. R. Kaye, "Analysis of a Distributed Control Loop for Data Transmission," Proc. of Symposium on Computer-Communications Networks and Teletraffic, Poly Institute Brooklyn, April 1972, pp. 47-58.

KLEI75 L. Kleinrock, Queueing Systems Volume I: Theory, Wiley-Interscience (New York), 1975.

KLEI76 L. Kleinrock, Queueing Systems Volume II: Computer Applications, Wiley-Interscience (New York), 1976.

KONH72 A. G. Konheim and B. Meister, "Service in a Loop System," Journal of the Association for Computing Machinery, 19, 1 (January 1972), pp. 92-108.

LABE77 J. Labetoulle et al., "A Homogeneous Computer Network," Computer Networks, North-Holland Publishing Co. (1977), pp. 225-240.

REFERENCES - SECTION 2 (Continued)

LIU77 M. T. Liu et al., "Traffic Analysis of the Distributed Loop Computer Network (DLCN)," Proc. of the 1977 National Telecommunication Conference, December 1977, pp. 31:5-1 - 31:5-7.

MAJI77 J. C. Majithia, "Analysis and Simulation of Message-Switched Loop Data Networks," Proc. IEE, 124, 3 (March 1977), pp. 193-197.

METC76 R. M. Metcalfe and D. R. Boggs, "Ethernet: Distributed Packet Switching for Local Computer Networks," Communications of the ACM Vol. 19, No. 7 (July 1976), pp. 395-404.

MUNT72 R. R. Muntz and F. Basket, "Open, Closed, and Mixed Networks of Queues with Different Classes of Customers," Technical Report No. 33, Stanford Electronic Laboratories, Stanford Univ., August 1972.

PARD77 R. Pardo, M. T. Liu, and G. A. Babic, "Distributed Services in Computer Networks: Designing the Distributed Loop Data Base System (DLDBS)," Proc. of the Computer Networking Symposium, December 1977, pp. 60-65.

PENN78 B. K. Penney and A. A. Baghdadi, "Survey of Computer Communication Loop Networks," Research Report 78/42, Dept. of Computing and Control, Imperial College of Science and Technology, London SW7, England.

PIER72a J. R. Pierce, "Network for Block Switching of Data," The Bell System Technical Journal, 51, 6 (July - August 1972).

PIER72b J. R. Pierce, "How Far Can Data Loops Go," IEEE Trans. on Comm., COM-20 (June 1972), pp. 527-530.

REAM75 C. C. Reames and M. T. Liu, "A Loop Network for Simultaneous Transmission of Variable-Length Messages," Proc. Second Annual Symposium on Computer Architecture, January 1975, pp. 7-12.

REAM76 C. C. Reames and M. T. Liu, "Design and Simulation of the Distributed Loop Computer Network (DLCN)," Proc. Third Annual Symposium on Computer Architecture, January 1976, pp. 124-129.

SHER70 D. N. Sherman, "Data Buffer Occupancy Statistics for Asynchronous Multiplexing of Data in Speech," Proc. IEEE ICC, 1970, Vol. 2, pp. 28-26 - 28-31.

SPRA72a J. D. Spragins, "Loops Used for Data Collection," Symposium on Computer-Communications Networks and Teletraffic, New York: Polytechnic Press, 1972, pp. 59-76.

SPRA72b J. D. Spragins, "Loop Transmission Systems - Mean Value Analysis," IEEE Transactions on Communications, COM-20, 3 (June 1972), pp. 592-602.

TROP78 C. Tropper, "Computerized Models for the Design and Analysis of Computer-Communications Systems," ESD-TR-79-128, Electronic Systems Division, AFSC, Hanscom AFB, MA, May 1979, ADA078022.

WEST72 L. P. West, "Loop-Transmission Control Structures," IEEE Transactions on Communications, COM-20, 3 (June 1972), pp. 531-539.

REFERENCES - SECTION 2 (Concluded)

WONG78 J. W. Wong et al., "Feasibility of a Loop System for Local Data Concentration," ICC 78, pp. 36.1.1 - 36.1.5.

YUEN72 M. L. T. Yuen et al., "Traffic Flow in a Distributed Loop Switching System," Proc. Symposium on Computer-Communications Networks and Teletraffic, New York: Polytechnic Press, 1972, pp. 29-46.

REFERENCES

SECTION 3. BUS NETWORKS

ABRA73 N. Abramson, "The ALOHA System," Computer Communication Networks, Abramson and Kuo, eds., Prentice-Hall, 1973.

AGRA77 A. K. Agrawala et al., "Analysis of an Ethernet-like Protocol," Proc. of the Computer Networking Symposium, NBS, December 1977, pp. 104-111.

ANDE75 G. A. Anderson and E. D. Jensen, "Computer Interconnection Structures: Taxonomy, Characteristics, and Examples," Computing Survey, 7, 4 (December 1975), pp. 197-213.

CHRI77 G. S. Christensen, "Data Trunk Contention in the HYPERchannel™ Network," Conference on a Second Look at Local Computer Networking, Univ. Computer Center, Univ. of Minn., October 1977.

COX62 D. Cox, Renewal Theory, Methuen (London), 1962.

DONN78a J. E. Donnelley and J. W. Yeh, "Simulation Studies of Round-Robin Contention in a Prioritized CSMA Broadcast Network," Third University of Minnesota Conference on Local Computer Networks, Minneapolis, Minnesota, 23-24 October 1978.

DONN78b J. E. Donnelley and J. W. Yeh, "Simulation of Round-Robin Contention in a Prioritized CSMA Network," Third University of Minnesota Conference on Local Computer Networks, Minneapolis, Minnesota, October 1978.

FUCH70 E. Fuchs and P. E. Jackson, ACM, Vol. 13, No. 12, 1970, pp. 752-757.

FRAN77 W. R. Franta and M. B. Bilodeau, "Analysis of a Prioritized CSMA Protocol Based on Staggered Delays," Technical Report 77-18, Computer Science Department of the University of Minnesota, September 1977.

KEND53 D. G. Kendall, "Stochastic Processes Occurring in the Theory of Queues and Their Analysis by the Method of the Imbedded Markov Chain," Annals of Mathematical Statistics, 24, 1953, pp. 338-354.

KLEI74 L. Kleinrock and F. A. Tobagi, "Carrier Sense Multiple Access for Packet Switched Radio Channels," in Proc. Int. Conf. Communications, Minneapolis, Minnesota, June 1974, pp. 21B-1 - 21B-7.

KLEI75a L. Kleinrock and S. Lam, "Packet Switching in a Multiaccess Broadcast Channel: Performance Evaluation," IEEE Transactions on Communications, COM-23, 4 (April 1975), pp. 410-421.

KLEI75b L. Kleinrock and F. A. Tobagi, "Packet Switching in Radio Channels: Part 1 - Carrier Sense Multiple-Access Modes and

Their Throughput-Delay Characteristics," IEEE Transactions on Communications, COM-23, 12 (December 1975).

KLEI75c L. Kleinrock, Queueing Systems, Volume I: Theory, Wiley-Interscience (New York), 1975.

KLEI76 L. Kleinrock, Queueing Systems, Volume II: Computer Applications, Wiley-Interscience (New York), 1976.

KLEI77a L. Kleinrock, "Performance of Distributed Multi-Access Comm. Systems," IFIP77, North-Holland Publishing Co., pp. 547-552.

KLEI77b L. Kleinrock and M. Scholl, "Packet Switching in Radio Channels: New Conflict-Free Multiple Access Schemes for a Small Number of Data Users," ICC '77 Conf. Record, Chicago, Illinois, 12-15 June 1977, pp. 22.1-105 - 22.1-11.

KLEI78 L. Kleinrock and Y. Yemini, "An Optimal Adaptive Scheme For Multiple Access Broadcast Communication," ICC '78, pp. 7.2.1-7.2.5.

KONH74 A. G. Konheim and B. Meister, "Waiting Lines and Times in a System with Polling," JACM, Vol. 21, No. 3, July 1974, pp. 470-490.

KONH76 A. G. Konheim, "Chaining in a Loop System," IEEE Transactions on Communications, COM-24 2 (February 1976), pp. 203-209.

REFERENCES - SECTION 3 (Continued)

LABA78 C. E. LaBarre, "Analytic and Simulation Results for CSMA Contention Protocols," ESD-TR-79-126, Electronic Systems Division, AFSC, Hanscom AFB, MA, May 1979, ADA070672.

LAM74 S. S. Lam, "Packet Switching in a Multi-Access Broadcast Channel with Application to Satellite Communication in a Computer Network," PhD Thesis, Rep. No. UCLA-ENG-7429, School of Engineering and Applied Sciences, UCLA, April 1974.

LAM75 S. S. Lam and L. Kleinrock, "Dynamic Control Schemes for a Packet Switched Multi-Access Broadcast Channel*," National Computer Conference, 1975, pp. 143-153.

LAM78 S. S. Lam, "An Analysis of the Reservation-ALOHA Protocol for Satellite Packet Switching," Int. Conf. Communications, pp. 27.3.1-27.3.5.

LUCZ78 E. C. Luczak, "Global Bus Computer Communication Techniques," Proceedings, Computer Networking Symposium, National Bureau of Standards, 1978, pp. 58-67.

MARK78 J. W. Mark, "Global Scheduling Approach to Conflict-Free Multiaccess via a Data Bus," IEEE Transactions on Communications, COM-26, 9 (September 1978), pp. 1342-1352.

MEIS77a N. B. Meisner et al., "Dual-Mode Slotted TDMA Digital Bus," Proc. Fifth Data Commun. Symp., 27-29 September 1977, pp. 5-14 - 5-18.

REFERENCES - SECTION 3 (Continued)

MEIS77b N. B. Meisner et al., "Time Division Digital Bus Techniques Implemented on Coaxial Cable," Proc. of Computer Networking Symposium, NBS, 1977, pp. 112-117.

METC73 R. M. Metcalfe, "Steady-State Analysis of a Slotted and Controlled ALOHA System with Blocking," Proc. Sixth Hawaii Conf. on System Science, January 1973, pp. 375-378.

METC76 R. M. Metcalfe and D. R. Boggs, "Ethernet: Distributed Packet Switching for Local Computer Networks," Communications of the ACM, Vol. 19, No. 7 (July 1976), pp. 395-404.

NAYL J. C. Naylor, "Data Bus Design Concepts, Issues and Prospects," EASCON.

NESS78 D. M. Nessett, "Protocols for Buffer-Space Allocation in CSMA Broadcast Networks with Intelligent Interfaces," Third Univ. of Minnesota Conference on Local Computer Networks, October 1978.

RAWS78 E. G. Rawson and R. M. Metcalfe, "FIBERNET: Multimode Optical Fibers for Local Computer Networks," IEEE Transactions on Communications, COM-26, 7 (July 1978), pp. 983-990.

ROBE73 L. G. Roberts, "Dynamic Allocation of Satellite Capacity Through Packet Reservation," AFIPS Conference Proc., National Computer Conference, 42, June 1973, pp. 711-716.

REFERENCES - SECTION 3 (Continued)

ROTH77 E. H. Rothauser and D. Wild, "MLMA - A Collision-Free Multi-Access Method," IFIP77, North-Holland Publishing Co., pp. 431-436.

SCHO76 M. Scholl, "Multiplexing Techniques for Data Transmission Over Packet Switched Radio Systems," PhD Thesis, Dept. of Computer Science, UCLA, September 1976.

SCHW77 M. Schwartz, "Computer-Communication Network Design and Analysis," Prentice-Hall, New Jersey, 1977.

THOR75 J. E. Thornton et al., "A New Approach to Network Storage Management," Computer Design, November 1975, pp. 81-85.

TOBA74 F. A. Tobagi, "Random Access Techniques for Data Transmission over Packet Switched Radio Networks," PhD Thesis, Computer Science Department, School of Engineering and Applied Sciences, UCLA, UCLA-Eng. 7499, December 1974.

TOBA76 F. A. Tobagi and L. Kleinrock, "Packet Switching in Radio Channels: Part III - Polling and (Dynamic) Split-Channel Reservation Multiple Access," IEEE Transactions on Communications, COM-24, 8 (August 1976), pp. 832-845.

REFERENCES - SECTION 3 (Concluded)

YEMI78 Y. Yemeni, "On Channel Sharing in Discrete-Time Packet Switched, Multiaccess Broadcast Communication," PhD Thesis, Dept. of Comp. Science, School of Engineering and Applied Sciences, UCLA, 1978.